UNIVERSITA' DEGLI STUDI DI MILANO
SCUOLA DI DOTTORATO IN INFORMATICA
DIPARTIMENTO DI SCIENZE DELL'INFORMAZIONE
CORSO DI DOTTORATO IN INFORMATICA - CICLO XXIII

TESI DI DOTTORATO DI RICERCA

A Rough Approach to Outlier Detection Problem in Spatio-Temporal Data

INF/01

TUTOR
Prof. Alfredo Petrosino
COORDINATORE DEL DOTTORATO
Prof. Ernesto Damiani

CANDIDATO
Alessia Albanese

Anno Accademico 2009-2010

Abstract

Spatio-temporal data mining is a growing research area dedicated to the development of algorithms and computational techniques for the analysis of large spatio-temporal databases and the disclosure of interesting and hidden knowledge in these data, mainly in terms of periodic hidden patterns and outlier detection. In this thesis, the attention has been focalized on outlier detection in spatio-temporal data. Indeed, detecting outliers which are grossly different from or inconsistent with remaining data is a major challenge in real-world knowledge discovery and data mining applications.
Nowadays, the high availability of data gathered from wireless sensor networks and telecommunication systems (such as GPS, GSM), that daily generate terabytes of data, has focalized the research attention on the interesting knowledge that can be gained from the analysis of spatio-temporal data. Spatio-temporal data are constituted by sampled locations at specific timestamps, tipically this kind of data deal with trajectory of moving objects that change their locations over time. The management and analysis of these data is interesting because undetected correlations between phenomena could be discovered and adequate improvements could be taken in many different fields, such as problem prevention, traffic management, discovery of meaningful behaviour pattern or accessibility of restricted areas and so on.

In this thesis, we face an unsupervised outlier detection problem in an unlabeled spatio-temporal data. Two main research contributions are reported in the following two main parts of this thesis.

In the first part of this thesis, we describe the first research contribution that consists of two non parametric methods. Most current methods for outlier detection give a binary classification of objects: is or is not an outlier or, but for many scenarios, it is more meaningful to assign to each object a degree of being an outlier (degree of outlier-

ness), that can be based on different rules, well known in literature. In both methods, the degree of outlierness of each object is based on the sum of the distances among the object itself and its k-nearest neighbours. The choice of developing a nearest neighbor based technique is that it is unsupervised in nature and does not make any assumptions regarding the generative distribution for the data. It is purely data driven. The former outlier detection method, called a two step approach, considers the spatial weight (component) in order to identify the spatial outliers, and, in a second time, considers also the temporal weight but only as a more refined level of anomaly detection. The latter outlier detection method, called ST-OutlierDetector, is a non parametric outlier detection approach that finds the top outliers in an unlabeled spatio-temporal data set. Our proposed method relies on a new fusion approach able to discover outliers according to the spatial and temporal features, at the same time: the user can decide the importance to give to both components (spatial and temporal) depending upon the kind of data to be analyzed and/or the kind of analysis to be performed.
Based on ST-OutlierDetector method, another contribution has been proposed. This contribution, the spatio-temporal outlierness degree map, is a visualization tool aimed at visualize the dataset structure with respect to the spatio-temporal outlier presence. It allows to make a 3D-plot (space and time) of the dataset by drawing them with different colors and also different color nuance based upon their outlierness degree. The map is built without setting, a-priori, the input parameter: outlier number to be found.

In the second part of this thesis, we describe the second research contribution that consists of a new outlier detection method, called ROSE (Rough Outlier Set Extraction). The attention has been focalized on outlier detection in spatio-temporal data using rough set theory. Most current methods for outlier detection exploit rough theory to define new rough weights as degree of outlierness. Our goal is representing the Outlier Set such as a Rough Outlier Set through its lower, upper approximation, remarking the benefits of keeping into account the objects belonging to the boundary. Moreover, we introduce a new set, called Kernel Set. This set is a selected subset of elements that is able to maintain the original data set both in terms of data structure and in terms of obtained results. In particular, we want to show the advantages of considering this new set. Indeed, we compare the Rough Outlier Set extracted by the entire data set (our Universe of the discourse) and the Rough Outlier Set extracted by the Kernel Set.

Contents

List of Figures

List of Tables

Acknowledgments

Firstly I would like to express my deep and sincere gratitude to my supervisor, Professor Alfredo Petrosino of University of Naples Parthenope. His understanding, encouraging as well personal guidance have provided a good basis for the present thesis.

During this work I have also collaborated with my colleagues of CVPRLab (Computer Vision and Pattern Recognition Laboratory) for whom I have great regard, and to whom I wish to extend my warmest thanks for helping me with my work.

A special thank goes to the referees for their useful comments on the manuscript.

I would like also to mention Professor Ernesto Damiani, Director of University of Milan's Ph.D. School in Computer Science and Dr. Lorena Sala of Ph.D. Secretariat for Her kindness and precious support.

I wish also to thank to my family for their unconditional encouragement.

Milan, March $25^{th}, 2011$

"Nature has perfection, in order to show that she is the image of God and defects, to show that she is only his image".

Blaise Pascal.

1 Introduction

This thesis is aimed at identify, efficiently, meaningful outliers in large unlabeled spatio-temporal datasets. In this first chapter, we begin by defining outlier detection as a significant task of data mining and by providing some motivation for our work. Then, the problem statement, the outline and the main contributions of the thesis are provided.

1.1 Data Mining

Many definitions of data mining have been provided: data mining has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" by Frawley et al. [34] while Hand and al. [41] define it as: "The science of extracting useful information from large data sets or databases". However, data mining is an umbrella term used with varied meaning in a wide range of contexts.

Data Mining is viewed as an interdisciplinary area focusing upon methodologies for extracting useful knowledge from data. The ongoing rapid growth of online data due to the Internet and the widespread use of databases have created an immense need for data mining methodologies. Data mining involves the use of sophisticated data analysis tools to nontrivial extraction of unknown knowledge such as valid patterns and relationships in large data sets. The adjective non trivial underlines that data mining tools are tasks more complex than the sql-queries. These tools include the use of statistical models, mathematical algorithms, and machine learning methods. Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction. Some of the most important technique are shown in Figure 1.2. Data mining is generally considered to be just one step in a larger process known as knowledge discovery in databases (KDD), a concept emerged in 1989 by Gregory Piatetsky-Shapiro to

refer to the broad process of finding knowledge in data. Other steps in the KDD process include steps of pre-processing, such as data cleaning and data transformation, before the core step (data mining), and steps of post-processing (interpretation and validation of the results) such as: pattern evaluation and knowledge presentation as shown in Figure 1.1. In order to attest the interest of scientific community and industry, annually, the

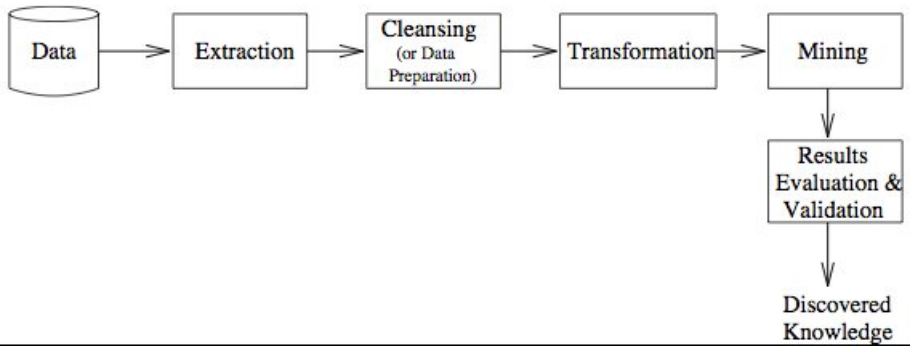


Figure 1.1: KDD Process

Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) holds a conference meeting to establish standards to define the parameters of the use of data mining tools. The Association is also responsible for assessing the ethical implications of the analysis of data from individuals and companies. A biannual journal is published by the group entitled SIGKDD Explorations.

The interest of data mining communities has attested also by the several commercial software for data mining, supercomputing data mining, text mining, and web mining. Moreover, some open-source projects have become an informal standard for defining data-mining processes. Just to name a few: Weka that stands for Waikato Environment for Knowledge Analysis, is free software available under the GNU General Public License. Weka is a popular suite of machine learning software written in Java, developed at the University of Waikato (New Zealand). RapidMiner, formerly YALE (Yet Another Learning Environment) is another open-source machine learning framework implemented in Java fully integrating Weka.

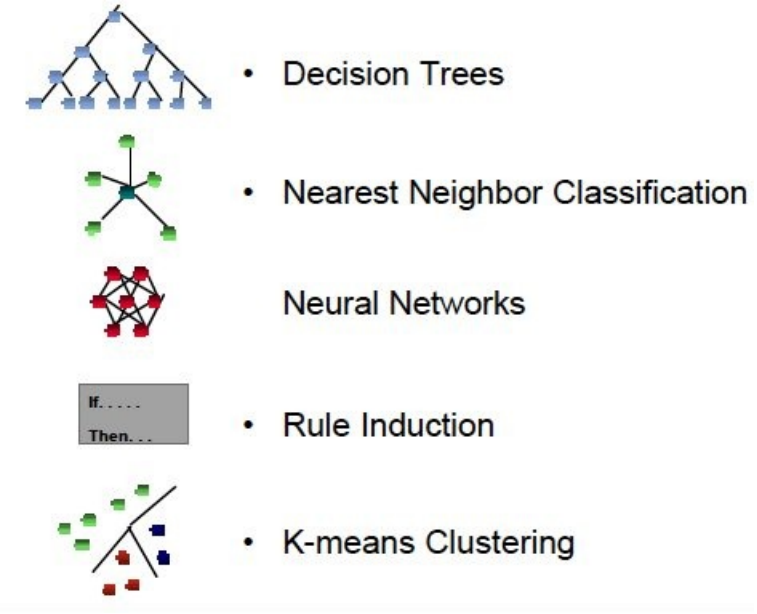


Figure 1.2: Data Mining Techniques (taken by Kurt Thearling - An Introduction to Data Mining)

1.1.1 Data Mining Application

Data mining has become increasingly common both in the public and private sectors. Organizations use data mining as a tool to survey customer information, reduce fraud and waste, and assist in medical research.
Several examples can be given; the insurance and banking industries use data mining applications to detect fraud and assist in risk assessment. Companies develop models that predict whether a customer is a good credit risk, or whether an accident claim may be fraudulent and should be further investigated, using customer data collected over several years.

The medical community sometimes uses data mining to help predict the effectiveness of a procedure or medicine. Pharmaceutical firms use data mining of chemical compounds and genetic material to help guide research on new treatments for diseases. In particular, in last years, this discipline has been widely used in the area of study on human genetics, in which, an important goal is to understand how the changes in an

individual's DNA sequence affect the risk of developing common diseases. This is very important to help improve the diagnosis and the prevention of the diseases.

Retailers can use information collected through affinity programs i.e., shoppers' club cards or frequent flyer points, to assess the effectiveness of product selection and placement decisions, coupon offers, and which products are often purchased together. This last is a data mining application, known as "market basket analysis", which received very much attention in the literature, in which retailers seek to understand the purchase behavior of customers. A legend tells that a famous supermarket chain did a study about customers' buying habits, discovering that beer and diapers were often purchased together. As a result of this, the retailers can decide to have the diapers next to the beer or to make a promotion involving just one of the two items, because it would likely drive to an increase in profit, rather than putting both items on promotion at the same time.

Private companies, such as telephone service providers, that have made a huge investment to acquire their customers, use data mining to create a "churn analysis", to assess which customers are likely to remain as subscribers and which ones have potential for defection but have not been contacted for retention purposes in recent times. Preventive actions can be followed for customers who have been identified as potential risky.

In the public sector, data mining applications were initially used as a means to detect fraud and waste, but they have grown also to be used for purposes such as measuring and improving program performance.

An important project, known as the National Security Analysis Center (NSAC), has the mission of bringing together "hundreds of millions of electronic records created or collected by the FBI and other government agencies" and of using that "vast ocean of data to predict who might be a potential terrorist, in the absence of intelligence linking the man or woman to any radical or extremist group" ([83]).

Moreover, in [83]: it has been reported that: "the federal government recovers millions of dollars in fraudulent medicare payments and the Justice Department has been able to assess crime patterns, by means of data mining tools".

Similarly, in [83] another example is in the aviation field: "data mining is used to review plane crash data to recognize common defects and recommend precautionary measures ".

Recently, particularly in United States, after the devastating events of 11 September 2001, data mining has been increasingly used in national security mission areas, to identify terrorist activities, and as crime-fighting technologies ([83]).

1.2 Outlier Detection

As early as 1620, Sir Francis Bacon wrote: "Whoever knows the ways of Nature will more easily notice her deviations; and, on the other hand, whoever knows her deviations will more accurately describe her ways". This mention attests that the awareness of outliers, in some form or another, has existed for at least several hundred years; also the awareness of the importance of studying and understanding the anomalies. The problem of outlier detection is a key problem in data mining. Let us introduce the concept of outliers.

1.2.1 Outlier Definitions

Coming across various definitions of an outlier, it seems that no universally accepted definition exists. Two classical definitions of an outlier include Hawkins and Barnett [42] and Lewis [16].
According to the former, "an outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism", where as the latter defines "an outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data".
The term "outlier" can generally be defined as an observation that is significantly different from the other values in a data set.
Outliers often occur due to the following reasons, which make occurrence of an outlier typically being an indication of an error (anomalies, noise) or an event, not conform to normal behavior, that may include interesting information to be further investigated. In the figure 1.3, taken by [26], a 2D plotting shows N_1 and N_2 two regions of normal behavior; points o_1 and o_2 are anomalies and points in region O_3 are also anomalies.

It is very critical to design an appropriate outlier detection approach for a given

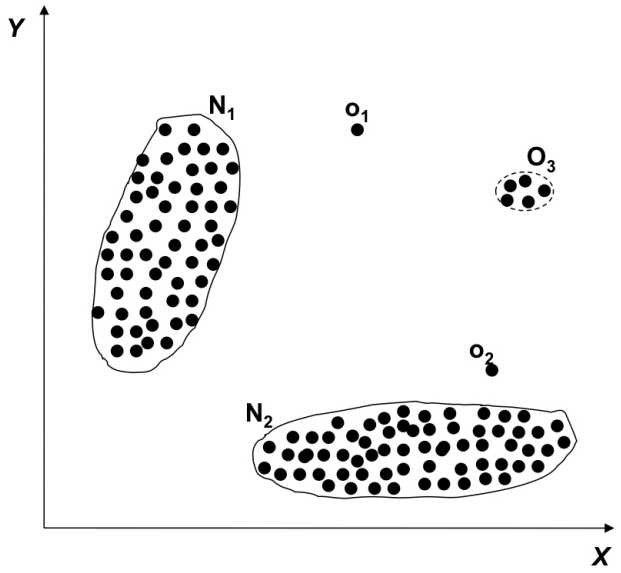


Figure 1.3: Synthetic Data set 2D

data set. There is no single universally applicable or generic outlier detection approach, specific application domains and type of data sets should be taken into account.

1.2.2 Outlier Detection as a Data Mining Task

While the field of data mining has been studied extensively, most of the work has concentrated on discovery of patterns. Outlier detection, as a branch of data mining, has many important applications and deserves more attention from data mining community. Most methods in the early work that detects outliers independently have been developed in field of statistics. Detecting and/or removing outliers is a very important task in data mining, for example error in large databases can be extremely common, so an important property of a data mining algorithm is robustness with respect to outliers in the database. Most sophisticated methods in data mining address this problem to some extent, but not fully, and can be improved by addressing the problem more directly.

1.2.3 Outlier Detection Applications

Over the years, outlier detection has been widely applied for the discovery of unexpected knowledge in different applications domains such as credit card fraud detection, discovering criminal behaviours, discovering computer intrusion, etc. As reported above, many interesting examples are reported below:

Fraud detection - Credit card transaction actually are the de facto standard per e-commerce. The growing number of transactions sometimes became an opportunity for thieves to steal credit card information in order to commit fraud. The credit card fraud detection domain presents a number of challenging issues for data mining and outlier detection: the detection and prediction of such buying pattern changes, could prevent thieves from fraud activity.

Intrusion detection - The task of anti-intruder detection represents one of the most important requirements in security network control of any critical infrastructure. Frequent attacks on computer systems may cause systems being disabled or completely collapsed. In this case, the identification of abnormal behaviour can find out malicious programs and identify unauthorized use, misuse by intruders with malicious intentions to computer network systems and keep out hackers.

Environmental monitoring - Many extreme weather events that occur in the natural environment such as a typhoon, hurricanes, drought and fire, often became a disaster for the human beings. The identification of certain atypical behaviors can accurately predict the probability of these phenomena and allow people to take appropriate measures on time.

Localization and tracking - Localization refers to the determination of the location of a set of objects. The collection of data can be used to localize the nodes of a network while simultaneously tracking a moving target. The data can be affected by errors, which make localization results not accurate. Detecting and removing these kind of abnormal data could improve the estimation of the location of objects and make tracking easier.

Logistics and transportation - Logistics is focused on the flow of materials and goods

from suppliers, through the organization and to the customers. Hence, it is essential to ensure product safety and product reliability issues during this process. Tracking and tracing infromation could find out exceptions such as, inappropriate quantity and/or quality of the product, and notify all trading partners in time.

1.2.4 Spatio-Temporal Data

Spatio-temporal data may arise in many contexts and areas like hydrology, ecology, geology, social sciences, brain imaging, wildlife population monitoring, tracking wild animals, tree defoliation in space and time, river flows, disease epidemic and also sociological and socio-economic phenomena. A particular application area of spatio-temporal data is in archaeology and palaeontology research that cannot be tackled readily using standard models because of the presence of uncertainty on both the temporal and the spatial scales. For example, the temporal information arises from chronometric dating methods, such as radiocarbon or uranium-series dating, which lead to estimated rather than exactly known calendar dates.

The spatio-temporal data sets are very large data set which are used to detect recognizable and meaningful patterns as well as to make predictions. In order to obtain a high degree of accuracy in analysis and predictions of a response variable, mathematical models are employed which explicitly include the underlying uncertainty in the data. Such models are statistical in nature and, if appropriately chosen, allow accurate forecasting in future time periods and interpolation over the entire spatial region of interest.

In order to model spatio-temporal data there is an obvious need to keep track of the spatial location, denoted by s in a region D, and the time point t. Different data types arise by the ways in which the points s are observed in D. Typical point reference data arise when s varies continuously over a fixed study region D. The set of spatial locations can either be fixed monitoring stations, like in an air pollution example, or can vary with time for example data obtained from a research ship measuring ocean characteristics as it moves about in the ocean. Discuss now two important and often used data types:

AREAL DATA The data are often called areal or block level data where the fixed region D is partitioned into a finite number of areal units with well defined boundaries, e.g. postcodes, counties or districts etc. Here an observation is thought to be associated with an areal unit of non-zero volume rather than a particular location point, e.g. a

latitude-longitude pair on the map. Typical areal data are represented by a choropleth map which uses shades of color or grey scale to classify values into a few broad classes, like a histogram. Such a map provides adjacency information of the areal units (blocks or regions). Some statistical issues here are spatio-temporal smoothing, inference and predictions for new areal units.

POINT DATA Spatial point pattern data arise when an event of interest, e.g. outbreak of a disease, occurs at random locations, that is, D is random and its index set gives the spatial point pattern; the notion of a response variable is not meaningful here, but there can be additional covariate information at the event locations. Spatio-temporal point are naturally found in a number of disciplines, including (human or veterinary) epidemiology where extensive data-sets are also becoming more common. One important distinction in practice is between processes defined as a discrete-time sequence of spatial point processes, or as a spatially and temporally continuous point process. On this second kind, the attention has been focalized. Tipically, this kind of data deal with trajectory of moving objects that change their locations over time. So, they are constituted by sampled locations at specific timestamps.

Spatio-Temporal Outlier Detection

Spatio-Temporal Outlier detection is an important research area due to the increasing amount of spatio-temporal data available and the need to understand and interpret them. Outlier detection refers to the problem of finding those patterns in data that do not conform to the expected behavior. Generally some techniques have been proposed for outlier detection in spatio-temporal data and overview of the research on spatio-temporal outlier detection. We can distinguish three main different categories: the first category considers that an outlier is a spatio-temporal outlier whose other attributes are significantly different from their spatial and temporal neighborhoods. In this category, there are works that deals with examining particular kind of data such as meteorological data and climatological data that describe natural phenomenon evolving in space and time. The second category is different from other approaches because it takes into account the influence of the underlying spatial objects that might be different at different spatial locations despite close proximity. This approach takes into consideration not only the spatial relationships but also the semantic relationships between spatial objects

and their respective areas of influence. The third category deals with flow anomaly; for particular kind of data coming from sensor networks, this category identifies, for pair of sensors, significantly mis-matched sensor readings (exceeding a given threshold) as a flow anomaly.

1.3 Thesis Contribution and Outline

In this section, the two main research contributions and the thesis outline have been reported.

1.3.1 Thesis Contribution

In the first major part of this thesis, we focus on outlier individuation problem in spatio-temporal data sets and we propose a non parametric approach, called ST-Outlier Detector; in the second major part of this thesis we focus our attention on the benefits coming from rough set theory applied to outlier detection in spatio-temporal data and we propose a rough set based approach called ROSE, that stands for Rough Outlier Set Extraction. Both our approaches belong to the first category.

A Non-Parametric Approach ST-Outlier Detector

Two distance-based outlier detection methods that find the top outliers in unlabeled spatio-temporal data sets are proposed. In contrast to the existing outlier detection methods that mainly consider only spatial component, the former proposed method is a two step approach that find spatial top outliers and spatio-temporal top outliers as a more refined level of anomaly detection; the latter proposed method is a combined approach that is able to discover outliers according to the spatial and temporal features, at the same time. The user can decide the importance to give to both components (spatial and temporal) depending upon the kind of data to be analyzed, by setting an input parameter. This approach has been already published [69].

A Rough-Set Approach ROSE

In contrast to the existing outlier detection methods that define the outlier set as a crisp set or as a ranked list of patterns, the proposed method rely on a rough set approach able to represent the outlier set according to the rough set approximations, i.e. as lower, upper approximations and relative boundary. The Rough Outlier Set Extraction (ROSE) manages the uncertainty of this kind of problems.
This approach is under review for publication.

1.3.2 Thesis Outline

In Chapter 1 and at various places throughout this thesis, we argue that outlier detection is a meaningful and important knowledge discovery task.
Chapter 2 describes aspects and characteristics of outlier detection problem giving the appropriate definitions.
Chapter 3 provides background and related works from the analysis of the existing literature on outlier detection problem. Particular attention has paid to outlier detection in spatio-temporal data and rough set-based techniques being two main relevant subjects of the following discussion.
In Chapter 4, we present two novel algorithms to identify outliers in spatio-temporal data. These algorithms have been tested on synthetic and real data sets.
In Chapter 5, we present a novel approach to identify outliers in spatio-temporal data from a rough set point of view.
In Chapter 6, we provide conclusions, possibilities for future work and a summary of this thesis.

2 Data Mining for Outlier Detection

In this chapter, aspects and characteristics of outlier detection problem have been described, giving the appropriate definitions.

2.1 Outlier Detection Problem

From a machine learning perspective, outlier detection can be categorized into a missing label problem or a one-class learning problem, depending on the way in which the normal samples are defined in a training data set [29] (see figure 2.1 taken by [26]).

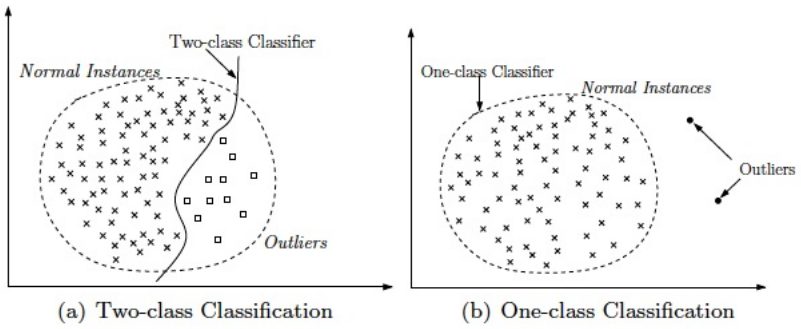


Figure 2.1: Comparison between missing label problem and one-class learning problem

2.1.1 Outlier Detection as Missing Label Problem

In a missing label problem, the data of interest consist of a mixture of normal samples and outliers, in which the labels are missing. The goal there is to identify outliers from the data and, in some applications, to predict outliers from an unseen data.

2.1.2 Outlier Detection as One-Class Learning Problem

In a one-class learning problem, normal samples are given as the training data. An outlier detector is built upon the normal samples to detect samples that deviate markedly from the normal samples, i.e., outliers. This is closely related to the standard supervised learning problem except that all the training samples have the same normal label.

2.2 Aspects of Outlier Detection Problem

It is very critical to design an appropriate outlier detection approach for a given data set. There is no single universally applicable or generic outlier detection approach, specific application domains and type of data sets should be taken into account. In Figure 2.2, taken by [26], a simple schema of the principal involved elements has been shown. The kind of available input and one of outlier required play an important role in designing a method (as described later), but also the specific application domain and semantic concepts influence the choices.

2.2.1 Nature of input data

A key component of any outlier detection technique is the nature of the input data. Input is generally a collection, called dataset, of data samples or data instances. Each data instance is described using a set of attributes, also named characteristics, features. The attributes can be of different types such as binary, categorical or continuous. In the simplest case, there is only one feature for each data instance (univariate) and otherwise multiple features (multivariate). The nature of attributes plays an important role in designing an outlier detection technique. For example, when applying statistical techniques, different statistical models have to be applied for continuous data and for categorical

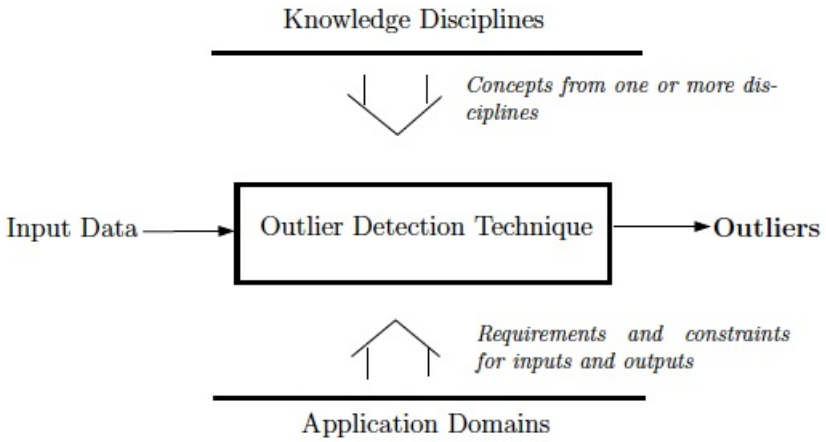


Figure 2.2: Anomaly Detection Process

data. Similarly, for techniques based on distance measures between data instances, the nature of attributes would determine the distance measure to be used. Input data can also be categorized based on the relationship present among data instances. Most of the existing outlier detection techniques deal with record data, in which no relationship is assumed among the data instances, even if, in some cases, such as spatial data, data instances can be related to each other.

2.2.2 Availability of supervision

As previously said, each data sample is described by a set of features and optionally could be associated by a class label to say that it belongs to a certain class. This is an important aspect: the availability of labeled data. In this case, we deal with supervised anomaly detection: labels are available for both normal data and anomalies.
Supervised approaches to anomaly detection have two major drawbacks: is not easy to obtain labelled data in many real-life applications, and moreover new types of rare events may not be included in the labelled data.
The category of unsupervised anomaly detection in which no labels are assumed, only take use of unlabeled data so, of course, are more general and based on the assumption

that anomalies are very rare compared to normal data.
Semi-supervised approaches to anomaly detection is when labels are available only for normal data. This kind of learning approaches improve the accuracy using supervision of some labeled data compared with that of unsupervised learning and, in the meantime, reduce the need for expensive labeled data which is required in supervised learning.

2.2.3 Type of anomaly: point, contextual, collective

The simplest type of anomaly, which is also the focus of the majority of research, is to detect an individual behavior instance that is considered as anomalous with respect to the rest of behaviors. This type of anomaly is called point anomaly. An example is shown in figure 2.3.

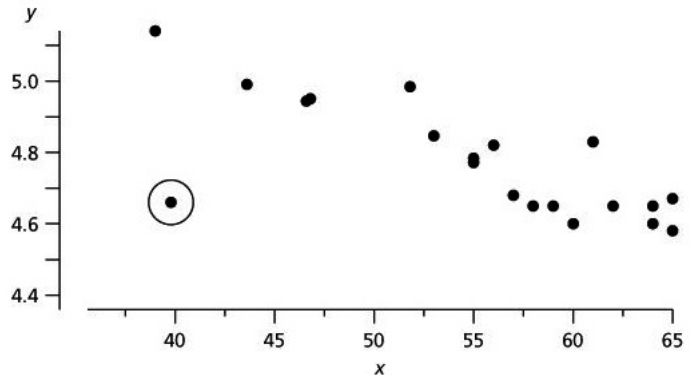


Figure 2.3: Example of point Anomaly

On the contrary, sometimes the individual behavior itself has similar features with others but it is anomalous in a specific context (e.g., neighborhood); then it is termed as a contextual anomaly. In the following figure 2.4 an example of contextual anomaly is shown, because the red highlighted values t_2 are normal values (the same of the green highlighted values t_1) but does not conform to the specified context (a such temperature value is normal is winter and may be not is summer. Another example of application of contextual anomaly is detecting individual anomalies in crowd scenes, i.e., human behaviors that are themselves normal but anomalous with respect to the rest of the

behaviors and so in a specific context.

In order to establish the context, it is necessary to distinguish between contextual and behavioral attributes. The contextual attributes are used to determine the context while the behavioral attributes define the non-contextual characteristics of an instance.

If a collection of related data instances is anomalous with respect to the entire data set,

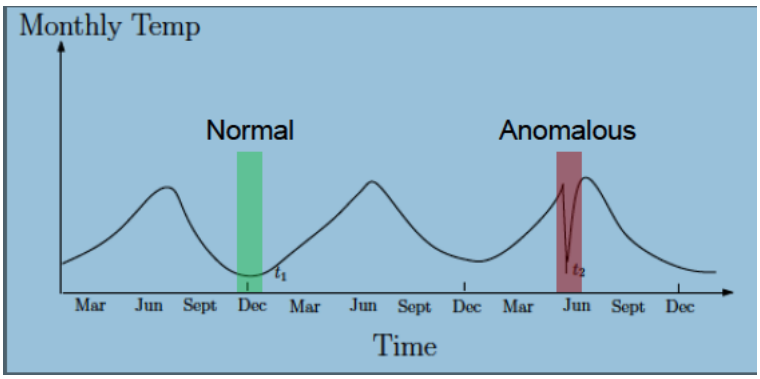


Figure 2.4: Example of contextual Anomaly

this is called a collective anomaly. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous. In the following figure 2.5 illustrates an example which shows a human electrocardiogram output with a collective anomaly highlighted in bold.

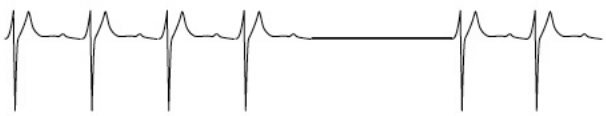


Figure 2.5: Example of collective Anomaly

2.2.4 Output of Outlier detection

An important aspect for any anomaly detection technique is the manner in which the anomalies are reported. Typically, the outputs, produced by anomaly detection techniques, are one of the following two types:
Labels: Label techniques assign a label (normal or anomalous) to each test instance.
Scores: Scoring techniques assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly. Thus, the output of such techniques is a ranked list of anomalies. An analyst may choose to either analyze the top few anomalies or use a cutoff threshold to select the anomalies.
Scoring-based anomaly detection techniques allow the analyst to use a domain–specific threshold to select the most relevant anomalies. Techniques that provide binary labels to the test instances do not directly allow the analysts to make such a choice, though this can be controlled indirectly through parameter choices within each technique.

2.3 Outlier Detection Taxonomy

Consider whether an outlier detection technique is suitable for a data set depending on several important aspects:

- the use of labelled data,
- the use of parameters of data distribution,
- the type and dimension of detected outliers,
- the degree of being outliers,
- the number of detected outliers at once.

We try now to schematize the elements that manage these choices:

- Use of labelled data
  - Supervised learning approach (labelled data)
  - UnSupervised learning approach (labelled data not necessary)
  - Semi-Supervised learning approach (labelled data only for training)

- Use of Parameters of data distribution
  - Parametric
    - Distribution-based techniques
    - Depth-based techniques
    - Graph-based techniques
  - Non-Parametric
    - Clustering-based techniques
    - Distance-based techniques
    - Density-based techniques
  - Semi-Parametric Method
    - Neural network-based techniques
    - Support vector machine-based techniques.

Supervised learning approaches employ a large amount of labeled data to train the model, but in practical learning scenarios, labeled data are expensive and difficult to be found, as they require the experienced human effort. Moreover, some rare events might not be included in labeled data. Unsupervised learning approaches are more general because the unlabeled data is relative easy to collect.

Parametric methods assume that the whole data can be modeled by one standard statistical distribution and then directly calculate the parameters of this distribution based on means and covariance of the original data. Hence, a point that deviates significantly from the data model is declared as an outlier. Non-parametric methods make no assumption on the statistic properties of data and instead identify outliers based on the fully dimensional distance measure between instances. Semi parametric methods map the data into a trained network model or a feature space to identify, as outliers, those instances that deviate from the trained network model or that are distant from other instances in the feature space, on the basis of some classification techniques.

Various outlier detection approaches work differently for different sets of data types. Based on the characteristics and attributes of data, data sets are divided into:

- simple
- complex

  - High dimensional
  - Mixed-type attributes
  - Sequence
  - Spatial
  - Streaming
  - Spatio-Temporal

where the complexity is referred to the semantic of data. Combining together a specific kind of dataset and a kind of approach we can obtain this subdivision:

- Simple Dataset
  - Parametric
    - Distribution-based technique
    - depth-based technique
    - graph-based technique.
  - Non-parametric
    - Clustering-based technique
    - Distance-based technique
    - Density-based technique.
  - Semi-parametric method
    - Neural network-based technique
    - Support vector machine-based technique.
- Complex Dataset
  - High dimensional
    - Distance-based technique
    - Subspace-based technique.
  - Mixed-type attributes
    - Graph-based technique.
  - Sequence

    - Clustering-based technique
    - Tree-based technique
  - Spatial
    - Distribution-based technique
    - Graph-based technique
  - Streaming
    - Graph-based technique
    - Model-based technique
    - Density-based technique
  - Spatio-Temporal
    - Clustering-based technique
    - Distance-based technique
    - Distribution-based technique

2.4 Summary

In this chapter, the general definition of Outlier Detection Problem from a machine learning point of view has been introduced and some different factors, that play a role in the specific formulation of the problem, such as the input data, the availability of labels as well as the specific application domain, have been analyzed. At the end, a taxonomy-like list, combining together the more common kind of data set and the various kind of approaches, has been provided.

3 Outlier Detection: Background and Related works

3.1 Introduction

Nowadays, the high availability of data gathered from wireless sensor networks and telecommunication systems, has focalized the research attention on the knowledge that can be gained from the analysis of a particular kind of data, spatio-temporal data. Moreover, new interesting research fields are coming up due to high availability of these data. The Moving Object Databases store geographical positions of moving objects at different times; these information typically represent moving object trajectories. In some application areas, such as GIS, computer vision, mobile computing and traffic analysis, huge amounts of data are generated and stored, explicitly or implicitly containing spatio-temporal information. Moreover, the proliferation of location-aware devices, such as wireless sensor networks or GPS devices, generate terabytes of data daily. These collections of spatio-temporal data contain interesting information and knowledge. The management and analysis of moving object trajectories is interesting because can provide benefits in many different fields: for example, problem prevention, discovery of meaningful behaviour pattern or accessibility of restricted areas and so on. In this way, undetected correlations between phenomena and rare event could be discovered so adequate improvements could be taken or new traffic policies could be defined to reduce traffic or the number of accidents. Here, the context is spatio-temporal data mining, i.e. a growing research area dedicated to the development of algorithms and computational techniques for the analysis of large spatio-temporal databases and the disclosure of interesting and

hidden knowledge in these data, in terms of periodic hidden patterns and outlier/novelty detection. Obviously, the topics of interest related to spatio-temporal data mining are several; the attention has been focalized on outlier detection in spatio-temporal data.
In this chapter, a brief overview of the most interesting outlier detection methods proposed in literature and then, in particular, for spatio-temporal data analysis, has been provided.

3.2 Outlier Detection Methods

In literature, the principal kinds of outlier detection approaches are the following:

1. Distribution-based approaches that use standard statistical distribution. They deploy some standard distribution model and recognize as outliers those points which deviate from the model. A large number of tests are required in order to decide which distribution model fits the arbitrary data set best. Fitting the data with standard distributions is quite costly.

2. Clustering-based approaches that have, as main objective, to discover clusters, and so they are not developed to detect outliers.

   Clustering is a technique aimed at grouping similar data instances in groups or clusters [47]. Although the main objective of clustering is to discover clusters, it has become an important tool for outlier detection and analysis. Indeed, several clustering-based outlier detection techniques have been developed. Most of these techniques rely on the key assumption that normal data instances belong to large and dense clusters, while outliers form very small clusters or are isolated elements.

3. Depth-based approaches are based on computational geometry and compute different layers of k-dimensional convex hulls. Outliers are more likely to be data objects with smaller depths. Depth-based approach is also applied for spatial outlier detection.

4. Distance-based methods use a distance metric to measure the distances among the data instances. Problems may occur if the parameters of the data are very different from each other in different regions of the data set.

5. Density-based approaches apply a local cluster criterion. Clusters are regarded as regions in the data space in which the objects are dense, and which are separated by regions of low object density (outlier). These regions may have an arbitrary shape and the objects inside a region may be arbitrarily distributed.

3.2.1 Distribution-based methods

Distribution-based methods rely on assumptions that the data follow a statistical distribution model e.g., Normal, Poisson, Binomial. Hence, a point that deviates significantly from the data model is declared as an outlier.
Distribution-based methods are the earliest parametric methods to face the outlier detection problem. As Parametric methods, they directly calculate the parameters of this distribution based on means and covariance of the original data. Then, they employ statistical tests to determine a point as an outlier depending on whether it deviates significantly from the data model [26].

Gaussian Model

This kind of technique assumes that the data is generated by a Gaussian distribution. The parameters are estimated using Maximum Likelihood Estimates (MLE). The distance of a data instance to the estimated mean is the anomaly score for that instance. A threshold is applied to the anomaly scores to determine the anomalies. Different techniques in this category calculate the distance to the mean and the threshold in different ways. A simple outlier detection technique (Shewhart [86]), is to declare all data instances that are more than 3σ distance away from the distribution mean μ, where σ is the standard deviation for the distribution. More sophisticated statistical tests have also been used to detect anomalies, as discussed in Barnett and Lewis [16], Barnett [15], and Beckman and Cook [18]. The most common outlier tests for normal distributions are: the Box-plot rule (Laurikkala et al. [56]), the Grubb's test (maximum normed residual test) used to detect anomalies in a univariate data set (Grubbs [39], Stefansky [89], Anscombe and Guttman [13]) and several other variants of Grubb's test, proposed to handle multivariate data sets (Aggarwal and Yu 2001 [6], 2008 [9]). Another variant of Grubb's test that uses the Mahalanobis distance is due to Laurikkala et al. [56], while another is due to Shekhar et al. [85] to handle graph structured data. The student's *t-test* has also been applied for anomaly detection in Surace and Worden [92] to detect damages in

structural beams. The multivariate version of *students't-test* called the Hotelling t^2-*test* is also used as an anomaly detection test statistic in Liu and Weng [58] to detect anomalies. Ye and Chen [108] use a χ^2 statistic to determine anomalies in operating system call data. The training phase assumes that the normal data has a multivariate normal distribution. Several other statistical anomaly detection techniques that assume that the data follows a Gaussian distribution have been proposed, but use other statistical tests, such as: Rosner test [78], Dixon test [38], Slippage Detection test [42], and so on.

Regression Model

Anomaly detection using regression has been extensively investigated for time-series data (Abraham and Chuang [2], Abraham and Box [1], Fox [33]). This kind of anomaly detection technique consists of two steps: in the first step, a regression model is fitted on the data, while in the second step, for each test instance, the magnitude of the residual (part of the instance which is not explained by the regression model) for the test instance is used to determine the anomaly score. A technique, called robust regression (Rousseeuw and Leroy [80]), solves the problem that the presence of outliers in the training data could influence the regression model parameters and consequently the result accuracy. A similar robust anomaly detection approach has been applied in Autoregressive Integrated Moving Average models (Bianco et al. [20], Ye and Chen [108]). Variants of the basic regression model-based technique have been proposed to handle multivariate time-series data (Tsay et al. [99]). Another variant that detects anomalies in multivariate time-series data generated by an Autoregressive Moving Average model, was proposed by Galeano et al. [36].

Mixture of Parametric Distributions

A kind of technique that uses a mixture of parametric statistical distributions to model the data. This category of techniques can be subdivided into two categories: the first subcategory models the normal instances and anomalies as separate parametric distributions, while the second sub-category of techniques models only the normal instances as a mixture of parametric distributions. For the first subcategory, the testing phase involves determining which distribution, normal or anomalous, the test instance belongs to. Abraham and Box [1] assume that the normal data is generated from a Gaussian distribution ($N(0, \sigma^2)$) and the anomalies are also generated from a Gaussian distribution with same mean but with larger variance. A test instance is tested using the Grubb's test on both distributions, and accordingly labeled as normal or anomalous. Similar techniques have been proposed in Lauer [57], Eskin [30], Abraham and Box [1], Box and

Tiao [23], and Agarwal [4]. The second subcategory of techniques models the normal instances as a mixture of parametric distributions. A test instance that does not belong to any of the learned models is declared to be an anomaly. Gaussian mixture models have been mostly used for such techniques Agarwal [5], to detect anomalies in mammographic image analysis (Spence et al. [88], Tarassenko [95]), and for network intrusion detection (Yamanishi and Takeuchi [106], Yamanishi et al. [107]. Similar techniques have been applied to detecting anomalies in biomedical signal data (Roberts and Tarassenko [74], Roberts 1999 [75] and 2002 [76]), where extreme value statistics (Extreme Value Theory - Pickands 1975) are used to determine if a test point is an anomaly with respect to the learned mixture of models or not. Byers and Raftery [25] use a mixture of Poisson distributions to model the normal data and then detect anomalies.

3.2.2 Depth-based methods

Outlier detection methods that are based on statistical depths have been studied in statistics and computational geometry. These methods provide a center-outward ordering of observations. Each data point is assigned by a depth [100] and outliers are expected to appear more likely in outer layers with small depth values than in inner layers with large depth values as shown in Figure 3.1. Depth–based methods are completely data–driven

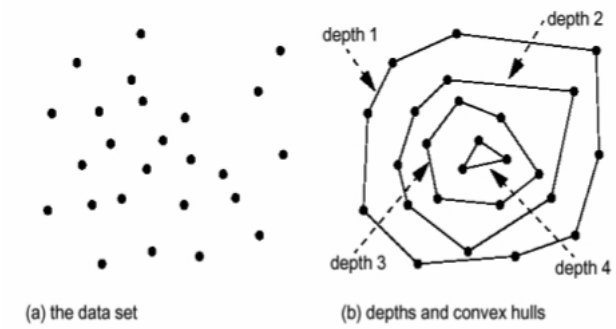


Figure 3.1: Figure taken by Preparata and Shamos 1988 [68]

and avoid strong distributional assumption. Moreover, they provide intuitive visualization of the data set via depth contours for a low–dimensional input space. Of the various

depths, spatial depth is especially appealing because of its computational efficiency and mathematical tractability. Spatial depth has been applied in clustering and classification problems [52], [37]. Because each observation from a data set contributes equally to the value of depth function, spatial depth takes a global view of the data set.

Rousseeuw and Leroy [80] describe two basic depth–based outlier detection techniques for low dimensional data sets, i.e., minimum volume ellipsoid (MVE) and convex peeling. MVE uses the smallest permissible ellipsoid volume to define a boundary around the majority of data and outliers are not in the densely populated normal boundary. Convex peeling maps data points into convex hull layers in data space according to peeling depth. Outliers are those points in the shallow convex hull layers with the lowest depth. Ruts and Rousseeuw [82] present an outlier detection approach using the concept of depth contour to compute the depth of points in a two–dimensional data set. Johnson et al. [51] propose a faster outlier detection approach based on computing two–dimensional depth contours in convex hull layers (Figure 3.2).

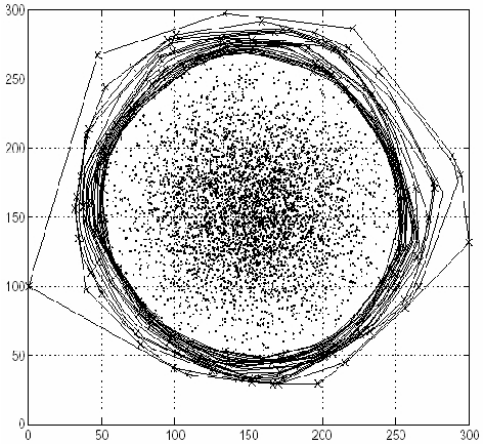
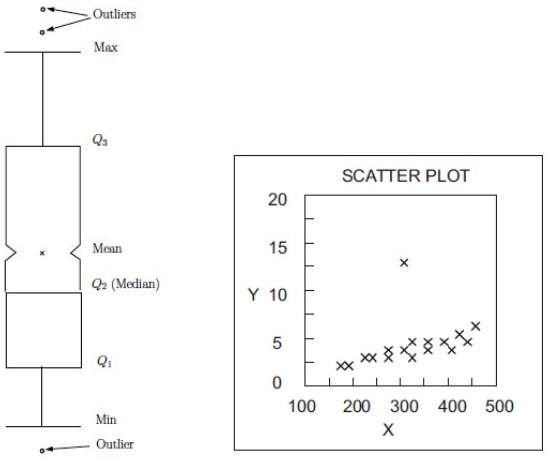


Figure 3.2: Figure taken by Johnson et al. 1998

3.2.3 Graph-based methods

Graph-based methods make use of a powerful tool data image and map the data into a graph to visualize the single or multi-dimensional data spaces. Outliers are those points

that are present in particular positions of the graph. These methods are suitable to identify outliers in real-valued and categorical data.
Laurikkala et al. [56] propose an outlier detection approach for univariate data based on box plot in figure 3.3 (a) which is a simple single-dimensional graphical representation. Using box plot, points that lie outside the lower and upper threshold are identified as



(a) (b)

Figure 3.3: An example: (a) Boxplot (b) Scatterplot

outliers. Also, these detected outliers can be ranked by the occurrence frequencies of outliers. Scatter plot [65] is a graphical technique to detect outliers in two-dimensional data sets (see figure 3.3 (b)). It reveals a basic linear relationship between the axis X and Y for most of the data. An outlier is defined as a data point that deviates significantly from a linear model. Moreover, spin plot [101] can be used for detecting outliers in 3-D data sets.

3.2.4 Clustering methods

Traditional clustering–based methods are developed to optimize the process of clustering of data, where outlier detection is only by–product of no interest. The novel clustering–based outlier detection methods can effectively identify outliers as points that do not

belong to clusters of a data set or as clusters that are significantly smaller than other clusters.
Clustering–based methods consider a cluster of small sizes, including the size of one observation, as clustered outliers. Some examples for such methods are the partitioning around medoids (PAM) and the clustering large applications (CLARA) [73]; a modified version of the latter for spatial outliers called CLARANS [72]; and a fractal dimension based method [14]. Note that since their main objective is clustering, these methods are not always optimized for outlier detection. In most cases, the outlier detection criteria are implicit and cannot easily be inferred from the clustering procedures. Other data mining algorithms in the literature find outliers as a side-product of clustering algorithms [7], [8], [10], [40]. However, these techniques define outliers as points which do not lie in clusters. Thus, the techniques implicitly define outliers as the background noise in which the clusters are embedded. A recent Density-Based Clustering and Outlier Detection algorithm (DBCOD in [109]) for discovering clusters and detecting outliers in a multidimensional database, solves clustering and outlier detection at the same time without losing the quality of clustering and outlier detection. It uses a novel concept called neighborhood–based local density factor (NLDF).

3.2.5 Distance-based methods

Distance-based methods are used to identify outliers based on the measure of full dimensional distance between a point and its nearest neighbors in a data set. Outliers are points that are distant from the neighbors in the data set. These methods generally define outliers based on a global view of the data set.
Knorr and Ng [54] introduced the notion of distance-based outliers, the $DB(p, d)$-*Outlier*.

Definition 1. *A data point x in a given data set is a $DB(p, d)$-Outlier if at least p fraction of the data points in the data set lies more than d distance away from x.*

The parameters p and d are to be specified by a user. So different choices of p and/or d lead to different observations being declared outliers. The authors of this definition proposed also some efficient algorithms for finding distance–based outliers. One algo-

rithm is a block nested–loop algorithm that has running time quadratic in the input size. Another algorithm is based on dividing the space into a uniform grid of cells and then using these cells to compute outliers. This algorithm is linear in the size of the database but exponential in the number of dimensions.
Outlier detection method based on Mahalanobis distance (MD) has been extensively studied in the statistics literature [77], [81], [79]. A fast algorithm provided by Rousseeuw and Van Driessen [81] makes robust version MD-based methods feasible for large sample size data. But the use of Euclidean rather than Mahalanobis distance speeds up the calculations considerably because computing and inverting covariance matrices, which are normally time consuming are not needed. However, relying exclusively on the Euclidean metric is equivalent to assuming that all variables are independent and have equal variances, a condition that is rarely observed in practice. Ignoring the dependence among variables will lead to inaccurate results in the majority of cases. Moreover, calculating the inter–point distances for all points in a dataset transforms an exploratory problem into a computational problem. The brute force method (exhaustive search) is clearly infeasible for most datasets, so algorithms have been proposed that are based on intelligent pruning.
Ramaswamy et al. [70] extended the notion of distance–based outliers by ranking each point on the basis of its distance to its *k-th nearest neighbor* and declaring the top n points as outliers.

Definition 2. *Given an input data set with N points, parameters n and k, a point p is D_n^k outlier if there are no more than n–1 other points p' such that $D^k(p') > D^k(p)$, where $D^k(p)$ is the distance between the object p and its k-th nearest neighbors.*

The authors of this definition develop a highly efficient partition–based algorithm for mining outliers. This algorithm first partitions the input data set into disjoint subsets, and then prunes entire partitions as soon as it is determined that cannot contain outliers.

Sun and Chawla [91] introduced a measure for spatial local outliers, which takes into account both spatial autocorrelation and spatially non-uniform variance of the data.
Angiulli et al. [11] designed a distance-based method to find outliers from a given data set and to predict if an unseen data point is an outlier based on a carefully selected

subset of the given data.
Aggarwal and Yu [6] investigated the influence of high dimensionality on distance-based outlier detection algorithms.
An analogous definition of outlier based on the *k-nearest neighbors* has been used in [31] for unsupervised anomaly detection to detect intrusions in unlabeled data. Data elements are mapped in a feature space and anomalies are detected by determining which points lie in sparse regions of the feature space.
More recently, Bay and Schwabacher [17], in order to find the top n distance based outliers of an input data set, augmented the naive distance–based nested loop algorithm, which finds the *k-nearest neighbors* of each data set point, with a simple pruning rule and randomization obtaining a near linear scaling on real, large, and high-dimensional data sets.
Ren et al. [73] present a faster way to implement the above outlier definition, utilizing their concept of P-trees, which examine the data 'vertically' rather than 'horizontally', that is, analyzing the data via its components rather than the individual observations.

3.2.6 Density-based methods

Density-based methods are proposed to take the local density into account when searching for outliers. These methods define outliers based on the local structure of the data set. Density-based algorithms: DBSCAN [32] is a widely known density-based clustering algorithm. The key idea in DBSCAN is that for each object in a cluster, the neighborhood of a given radius ε has to contain at least a minimum number $MinPts$ of objects, where ε and $MinPts$ are input parameters. GDBSCAN [84] extends the famous algorithm DBSCAN to apply to spatial database. OPTICS [12] has been devised to reduce the burden of determining parameter values in DBSCAN. IDBSCAN is an improved sampling-based DBSCAN which can cluster large-scale spatial databases effectively. Since DBSCAN uses global parameters, it can not distinguish small, close and dense clusters from large and sparse clusters.
To solve this problem, a neighborhood based clustering algorithm named NBC [114] is proposed. It uses the neighborhood relationship among objects to build a neighborhood based clustering model to discover clusters.
LOF [24] is a representative density-based outlier detection algorithm. An outlier is de-

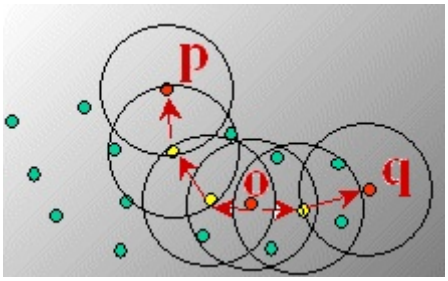


Figure 3.4: DBSCAN

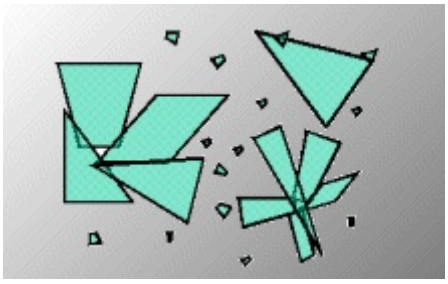


Figure 3.5: GDBSCAN

fined using the *local outlier factor* (LOF) of the current object, which depends on the local density of its neighborhood. LOF assign to each object a degree of being an outlier, objects with high LOF value are detected as outlier. Unfortunately, the work done in [24] requires the computation of LOF value for all objects which is rather expensive because it requires a large number of *k-nearest neighbors query*.
Similar to LOF, Zhang et al. [111] propose two novel algorithms LDBOD and LDBOD+ for outlier detection from the viewpoint of local distribution, which is characterized through three proposed measurements, local-average-distance, local-density, and local-asymmetry-degree. Many other extensions to LOF have been proposed, as aLOCI [66], Local Distance–based Outlier Factor (LDOF) for scattered real-world datasets [112]. Tang et al. [94] present an outlier detection approach based on a *connectivity–based outlier factor* (COF) that results more effective, especially for sparse data sets. The degree of outlierness COF is calculated using the ratio of the average distance from the point

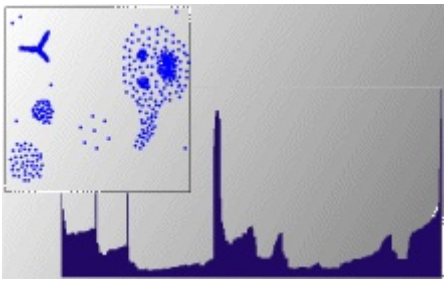


Figure 3.6: OPTICS

to its k–distance neighbors and the average distance from its k–distance neighbors to their own k–distance neighbors. Points that have the largest COF values are declared as outliers.

3.2.7 Classification techniques

Some classification techniques have been applied to outlier detection.
Classification-based anomaly detection techniques operate in a two-phase fashion. The training phase learns a classifier using the available labeled training data. The testing phase classifies a test instance as normal or anomalous, using the classifier.

Neural networks based methods can autonomously model the underlying data distribution and distinguish between the normal and abnormal classes. Those data points that are not reproduced well at the output layer are considered as outliers.
A basic multi-class anomaly detection technique using neural networks operates in two steps. First step: a neural network is trained on the normal training data to learn the different normal classes, second step: each test instance is provided as an input to the neural network. If the network accepts the test input, it is normal and if the network rejects a test input, it is an anomaly [90], [62]. Several variants of the basic neural network technique have been proposed in literature that use different types of neural networks. Replicator Neural Networks 3.7 have been used for one-class anomaly detection [43], [104]. A multi-layer feed forward neural network is constructed that has the same number of input and output neurons (corresponding to the features in the data). Also Support

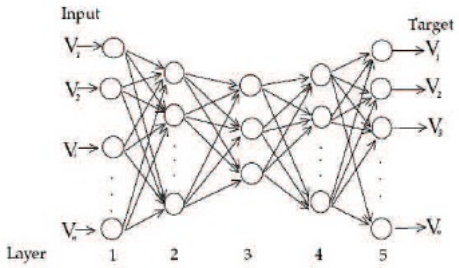


Figure 3.7: Replicator Neural Networks

Vector Machines [102] have been applied to anomaly detection in the one-class setting. Such techniques use one class learning techniques for SVM [71] and learn a region that contains the training data instances (a boundary). Kernels, such as radial basis function (RBF) kernel, can be used to learn complex regions. For each test instance, the basic technique determines if the test instance falls within the learned region. If a test instance falls within the learned region, it is declared as normal, else it is declared as anomalous. The basic technique has also been extended to detect anomalies in several fields, i.e. in temporal sequences [59], [60]. A variant of the basic technique [96], [97], [98] finds the smallest hypersphere in the kernel space that contains all training instances, and then determines on which side of that hypersphere a test instance lies. If a test instance lies outside the hypersphere, it is declared to be anomalous. Song et al. [87] use Robust Support Vector Machines (RSVM), which are robust to the presence of anomalies in the training data.

3.2.8 Other techniques

Being concerned with the complex data sets, several novel outlier detection methods have been proposed to deal with data with specific semantic.

Subspace-based methods project the data into a low-dimensional subspace and declare a point as an outlier if this point lies in an abnormal lower-dimensional projection, where the density of the data is exceptionally lower than the average. These methods reduce

the dimensions of data and efficiently identify outliers in high dimensional data sets [6] .

Tree-based methods construct a specific tree as index to decompose data structure and use an efficient similarity measure for the sequence data to distinguish outliers from non-outliers. These methods efficiently identify outliers only by examining nodes near the root of tree [91].

Model-based methods detect outliers by the construction of a model, which can represent the statistical behaviour of data stream. Outliers are those points that deviate significantly from the learned model. These methods can efficiently deal with the streaming data in an online fashion [44].

3.3 Outlier Detection Methods on spatio-temporal data

Most existing spatio-temporal outlier detection techniques focus on detecting spatial outliers, which only considers the spatial attributes of data or the spatial relationships among neighbors. However, in all geographic phenomena evolving over time, temporal aspects and spatio-temporal relationships existing among spatial data points also need to be considered in detecting outliers. Currently, some works have addressed the detection of spatio-temporal outliers in data sets on the basis of clustering concepts and statistical tests.
Cheng and Li [27] introduce a formal definition of ST (spatio-temporal) outliers:

Definition 3. *A spatio-temporal object (ST-***Outlier***) whose thematic attribute values are significantly different from those of other spatially and temporally referenced objects in its spatial or/and temporal neighborhoods.*

The definition indicates that ST-outliers are identified by comparing the spatio-temporal points with their spatio-temporal neighbors. Considering the temporal aspects, the authors declare a point as a ST-Outlier by checking if the point's attribute value at time T is significantly different from the statistical attribute values of its neighbors at time

T-1 and T+1.
Cheng and Li [27] further propose a four-step approach to detect ST-outliers, i.e., classification, aggregation, comparison and verification. In particular, the classification step aims at finding out the spatio-temporal points of interest by clustering the input data, which can be achieved by either supervised classification based on priori knowledge of the data or unsupervised classification if prior knowledge of data is not available. The aggregation step uses different spatial scales of the data to generate different clusters and effectively filter the noises. In comparison step, potential spatial outliers can be identified by comparing the results obtained from the classification step with the results obtained from the aggregation step. The verification step further compares these potential spatial outliers with their temporal neighbors in a continuous pattern. If the difference value is greater than a statistical threshold, these outliers will be considered as true ST-Outliers. Derya Birant, Alp Kut [21] define a similar definition of ST-Outlier as [27]:

Definition 4. *A Spatial Outlier (S-***Outlier***) is an object whose non-spatial attribute value is significantly different from the values of its spatial neighbors.*

Definition 5. *A Temporal Outlier (T-***Outlier***) is an object whose non-spatial attribute value is significantly different from those of other objects in its temporal neighborhood.*

Derya Birant, Alp Kut [21] present a ST-outlier detection approach based on clustering concepts called ST-DBSCAN. In particular, this approach consists of three steps, clustering, checking spatial neighbors, and checking temporal neighbors. In the clustering step, an efficient clustering technique DBSCAN [32] has been improved in supporting temporal aspects and detecting outliers in clusters with different densities. As a result, potential outliers are those points which do not belong to any of clusters. The following two steps further verify these potential outliers. In the checking spatial neighbors step, a potential outlier is labelled as a spatial outlier if its statistic value is outside a user-specified confidence interval. In the checking temporal neighbors step, if this spatial outlier is significantly different from its temporal neighbors in consecutive time units, it is labelled as a true ST-outlier. This approach uses several pre-defined parameters and some of them are very sensitive for the performance of outlier detection.

Wu, et al. [105] propose a spatio-temporal outlier detection algorithm called Outstretch, which discovers the outlier movement patterns of the top-k spatial outliers over several time periods. The top-k spatial outliers are found using the *Exact-Grid Top-k* and *Approx-Grid Top-k* algorithms, which are an extension of algorithms developed by Aggarwal et al.

3.4 RST-based Outlier Detection Methods

Granular Computing and Rough Set theory provide excellent methods and frameworks for Outlier Detection tasks.
Let us introduce the main aspects of classical rough set theory.

3.4.1 Rough Set Theory

Rough-set theory is a new and highly accepted paradigm that is used to deal with uncertainty, vagueness, and incompleteness. Rough set theory, proposed by Zdzislaw Pawlak [67], is a model of approximate reasoning. The main idea is based on the indiscernibility relation that describes indistinguishability of objects. Concepts are represented by lower and upper approximations. In applications, rough set methodology focuses on approximate representation of knowledge derivable from data. Rough Set Theory (RST) can be approached as an extension of the Classical Set Theory, for use when representing incomplete knowledge. The theory of rough sets begins with the notion of an approximation space, which is a pair $\langle U, R\rangle$, where U is a non–empty set, called the universe of discourse, and R is an equivalence relation on U, i.e., R is reflexive, symmetric and transitive. The relation R decomposes the set U into disjoint classes in such a way that two elements x and y are in the same class if and only if $(iff)(x, y) \in R$.
Let U/R denote the quotient set of U by the relation R, and

$$U/R = \{X_1, X_2, ..., X_m\}$$

where X_i is an equivalence class of R, $i = 1, 2, ..., m$. If the two elements x and y in U belong to the same equivalence class $X_i \in U/R$, then we can say that x and y are indistinguishable. The equivalence classes of R and the empty set $\emptyset$ are the elementary sets in the approximation space $\langle U, R\rangle$. Given an arbitrary set $X \in \{2^U\}$, in general, it

may not be possible to precisely describe $X \in \langle U, R\rangle$. X can be characterized by a pair of lower and upper approximations defined as

$$\underline{R}(X) = \bigcup_{X_i \subseteq X} X_i \text{ and } \overline{R}(X) = \bigcup_{X_i \cap X \neq \emptyset} X_i.$$

That is, the lower approximation $\underline{R}(X)$ is the union of all the elementary sets which are subsets of X, and the upper approximation $\overline{R}(X)$ is the union of all the elementary sets which have a non–empty intersection with X. The interval $[\underline{R}(X), \overline{R}(X)]$ is the representation of an ordinary set X in the approximation space $\langle U, R\rangle$ or is simply called the rough set of X. The lower (respectively, upper) approximation $\underline{R}(X)$ [respectively, $\overline{R}(X)$] is interpreted as the collection of those elements of U, which definitely (respectively, possibly) belong to X. Furthermore, a set of X is said to be definable (or exact) in $\langle U, R\rangle$ iff $\underline{R}(X) = \overline{R}(X)$. Two numerical characterizations of imprecision of a subset X in the approximation space $\langle U, R\rangle$ ha been defined: accuracy and roughness.
Accuracy of X, which is denoted by $\alpha_R(X)$, is the ratio of the number of objects on its lower approximation to that on its upper approximation, namely

$$\alpha_R(X) = \frac{|\underline{R}(X)|}{|\overline{R}(X)|}.$$

The roughness of X, which is denoted by $\rho_R(X)$, is defined as $\rho_R(X) = 1 - \alpha_R(X)$. Note that the lower the roughness of a subset, the better is its approximation. Furthermore, the following conditions are valid.

1. As $\underline{R}(X) \subseteq X \subseteq \overline{R}(X),\ 0 \leq \rho_R(X) \leq 1$
2. By convention, when $X = \emptyset,\ \underline{R}(X) = \overline{R}(X) = \emptyset$ and $\rho_R(X) = 0$
3. $\rho_R(X) = 0$ iff X is definable in $\langle U, R\rangle$

3.4.2 Outlier Detections Methods using Rough Set Theory

Granular Computing and Rough Set theories seem to provide excellent methods and frameworks for such tasks. Some outlier detection techniques, that are exploring this area, are reported in the following. Nguyen in [61] discusses methods for the detection and evaluation of outliers, as well as how to elicit background domain knowledge from outliers using multi-level approximate reasoning schemes.

Y. Chen, D. Miao, and R. Wang in [28] demonstrate the application of granular computing model using information tables for the outlier detection. They proposes a novel definition of outliers - GrC (granular computing)- based outliers. A definition of granular outlier factor (GOF) based on the distance between granules is given, which can indicate the degree of outlierness for every granule in the granular computing model. An algorithm to find such outliers is also given.
F. Jiang, Y. Sui and C. Cao in [49] propose a new definition for outliers in rough set theory which exploits the rough membership function. In this approach, similar to Breunig's method LOF, a rough outlier factor (ROF), which indicates the degree of outlierness for every object with respect to a given subset of universe is defined. An algorithm to find such outliers in rough set theory is also given.

3.4.3 Spatio-temporal data using Rough Set

Bittner in [22] represents spatio temporal data using rough set. Spatio-temporal regions are defined as pairs consisting of a spatial and a temporal component and topological relations between them are also defined. Using the notion of rough sets, Bittner defines approximations of spatio-temporal regions and relations between those approximations. Based on relations between approximated spatio-temporal regions, configurations of spatio-temporal objects can be characterized even if only approximate descriptions of the objects forming them are available.

3.5 Summary

An overview of outlier detection methods and in particular outlier detection methods in spatio-temporal data has been provided. A quick look to rough set based outlier detection methods has also been provided.

4 A Non Parametric Approach: ST-Oulier Detector

4.1 Introduction

In this chapter, we are going to introduce two new non parametric approaches: a two step approach and ST-Oulier Detector.

At this aim, the outlier detection problem can be defined as follows: given a set of N data objects and an expected number of outliers, n, find the top n objects that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data. One of the most popular kind of approaches for detecting outliers is the distance-based approach, in which the distance of a object from its k nearest neighbors is calculated. The rationale behind this approaches is that: if the neighboring objects are relatively close, then the point is considered normal; otherwise, if the neighboring objects are far away, then the object is considered outlier. The advantages of this kind of approaches are that no explicit distribution needs to be defined to detect outliers and can be applied to any feature space for which a distance measure can be defined. It is an approach unsupervised in nature and purely data driven.

Given a distance measure on a feature space, there are many different definitions for the distance-based outliers, we mention two of them that are more relevant for our work. One is due to Ramaswami et al. [70], and other, most recently, is due to Angiulli and Pizzuti [11]. Given a k and n, an object p is an outlier if no more than n-1 other objects in the dataset have a higher value for D_k than p where D_k is a degree of outlierness, computed on the basis of the full distances among the object p and its *k-nearest neighbors*.

This means that the top n points, having the maximum D_k values, are considered outliers.
For Angiulli and Pizzuti, the sum of the distances from the object itself and its k-*nearest neighbors* is assigned, as a weight, to each object of the dataset. In this way, each object is assigned by a degree of outlierness, accordingly to outlier detection scoring techniques. The disadvantage of distance-based approaches is its high computational complexity. The computational complexity is directly proportional to both the dimensionality of the data and the number of objects. In this direction, many different pruning strategies have been proposed in literature to find these outliers efficiently.

4.2 The proposed solutions

With recent advances in sensory and mobile computing technology, enormous amounts of data about moving objects are being collected. With such data, it becomes possible to automatically identify suspicious behavior in object movements. Anomaly detection in massive moving objects has many important applications, especially in surveillance and homeland security. Due to the sheer volume of spatiotemporal data associated with moving objects, it is challenging to develop a method that can efficiently and effectively detect anomalies of object movements in complex scenarios. The problem is further complicated by the fact that anomalies may occur at various levels of abstraction and be associated with different time and location granularities. In this chapter, we analyze the problem of anomaly detection in moving objects and propose an efficient and scalable non parametric method, called *ST*-Outlier Detector, which faces an unsupervised anomaly/outlier detection in spatio-temporal data. Accordingly to the taxonomy shown in the previous chapter, a distance/clustering approach has been chosen.

Our prospected solution is a non parametric approach because we make no assumption about the underlying distribution. In particular, we search for neighbors based on full distances. Shall we start to explain the two phases approach that has been the starting point of our research.

4.2.1 An initial two phases approach

Many applications track the movement of mobile objects, which can be represented as sequences of time-stamped locations. The movement of an object is tracked as a n-length sequence S of spatial locations, one for each timestamp in the history (the locations are sampled over a long history), of the form:

$$D = \{(\mathbf{l}_0, t_0), (\mathbf{l}_1, t_1), ..., (\mathbf{l}_{N-1}, t_{N-1})\}$$

where l_i is the object location (expressed in terms of spatial coordinates) at time t_i.

Given a long history of such spatio-temporal data D and a distance measure, the solution proposed here consists in a two-step approach to detect spatio-temporal outliers in large databases:

- the first phase is spatial outlier detection,
- the second phase is temporal outlier detection as a more refined level of anomaly detection.

The solution uses a well-known distance-based approach, using Euclidean distance; in particular, the outlier definition is based on the k-nearest neighbors. The idea is as follows: assign to each point a weight based on the sum of all the distances between the point itself and some k (input parameter) nearest neighbors, before among a small outlier candidate set and then, iteration by iteration, in a more precise way, therefore:

$$\omega_k(p, D) = \sum_i^k dist(p, nn_i(p, D)),\ \forall p \in D$$

where $\omega_k(p, D)$ is the weight of p with regard to k in D, $nn_i(p, D)$ is the *i–th nearest neighbor* of p in D, *dist* is the Euclidean distance and D is the original dataset. In this context, the outlier detection problem can be formalized as follows: find out n points that score the greatest weights. The result set will be:

$$\{S_{1,k}, S_{2,k}, ..., S_{n,k}\}$$

where $S_{i,k}$ is the object having the *i-th* greatest weights with regard to k, $i = 1, ..., n$ and n represents the number of top outliers required.
Intuitively, the notion of weight captures the degree of dissimilarity of a point or, more in general, of an object with respect to its neighbors. The weight is used to assign a degree of outlierness to each point, outliers are those points that have the highest ranks.

This definition has been used in several works on outlier detection topic.
Now, we want to apply it to spatio-temporal context.
The first phase works only on spatial components. The second phase works on temporal attributes (timestamp), using again a distance-based approach in order to refine the solution set obtained in the previous phase. The aim is to use also temporal features in order to find out two separate subsets: spatial outliers subset and spatio-temporal outliers subset. Firstly, it is necessary to fix a temporal criterion that allows to establish who is the temporal neighbor of another point. For example, two events can be defined "near in time" if they happen every day at the same time. The Euclidean distance is used two times to calculate two different weights: the former computation takes into account spatial values (cartesian coordinates x_i and y_i) and the latter uses non-spatial values, i.e. temporal components t_i. Given a point p, the Euclidean distance on temporal component, indicated by $dist_t$, will be based on the chosen criterion, such as:

$$dist_t(p, nn_i(p, D)) = |t - t_i| \quad \forall i = 1, ..., k$$

where t and t_i are temporal attributes of p and of $nn_i(p, D)$ respectively. In this way, the temporal value of an object is compared with the temporal values of spatial neighbors in order to find out the spatio-temporal neighbors (such as previous day, next day in the same year or the same day in other years and so on). Now, for each point, the nearest spatial k-neighbors will be iteratively picked up, and among these, also the temporal component will be checked and the spatial neighbors will be labelled as nearest temporal neighbor or not in case of the criterion will be respected or not. At the end of this process, for each outlier a structure will keep the k nearest spatial neighbors and eventually also spatio-temporal neighbors.
The rationale behind this approach is that a point is a spatio-temporal outlier whenever none among its k spatial nearest neighbors is also a temporal neighbor; and, vice versa, if a point owns temporal neighbors it will be only a spatial outlier.

4.2.2 A combined approach

Let us consider the movement of an object as a N-length sequence

$$D = \{(\mathbf{l}_0, t_0), (\mathbf{l}_1, t_1), ..., (\mathbf{l}_{N-1}, t_{N-1})\} \tag{4.1}$$

where l_i is the object location (expressed in terms of spatial coordinates) at time t_i. This assumpion fully agrees with many applications that track the movement of mobile objects, represented as sequences of time-stamped locations. Our non-parametric approach, called *ST*-Outlier Detector Algorithm (Spatio-Temporal Outlier Detector Algorithm), relies on the consideration that, in the parametric approaches, the choice of distribution parameters, to be estimated, is not always a simple task because of the poor knowledge about the data to be analyzed.
The rationale is to use the relative location of an object to its neighbours to determine the degree to which the object deviates from its neighbourhood.
The proposed outlier detection algorithm is characterized by two main contributions:

1. it allows to find the spatio-temporal outliers in a combined way: each point will have an unique attribute depending both from spatial distance measure and temporal distance one in an user stated percentage.

2. it allows to find only the spatial outliers or the temporal ones (limit cases).

The approach presented faces the spatio-temporal outlier detection problem from a new perspective, that is considering a mixture of spatial and temporal features. Let us consider that dataset features are only space and time, some definitions have to be provided:

Definition 6. *A Spatial Outlier (**S-Outlier**) is an object whose spatial attribute value is significantly different from those of its closer objects.*

Definition 7. *A Temporal Outlier (**T-Outlier**) is an object whose temporal attribute value is significantly different from those of its closer objects.*

The definition 6 states that a spatial outlier has no objects or a small group of objects in its spatial neighborhood. The definition 7 states that a temporal outlier has no objects or a small group of objects in its temporal neighborhood. According to them, a Spatio-Temporal Outlier (**ST-Outlier**) is an object which respects both the definitions 6 and 7 above.

Given a dataset D as in 4.1 and a distance measure *dist*, the proposed solution adopts a distance-based approach and, in particular, the outlier definition based on the k-nearest

neighbors (KNN) method. The main idea of this method is to assign to each point a weight based on the sum of all the distances between the point itself and the k (input parameter) nearest neighbors

$$\omega_k(p, D) = \sum_{i}^{k} dist(p, nn_i(p, D)) \quad \forall p \in D \tag{4.2}$$

where $\omega_k(p, D)$ is the weight of p with regard to k in D, $nn_i(p, D)$ is the *i–th nearest neighbor* of p in D, *dist* is the Euclidean distance and D is the original dataset. The outlier detection problem can be formalized as follows: find the set of n objects that score the greater weights. The outlier set, the result set, is:

$$O = \{S_{1,k}, S_{2,k}, \ldots, S_{n,k}\} \tag{4.3}$$

where $S_{i,k}$ is the object having the *i-th* greatest weight with respect to k, $i = 1, \ldots, n$ and n represents the number of outliers required.

Intuitively, the notion of weight captures the degree of dissimilarity of an object with respect to its neighbors and hence outliers are those objects that have the largest weights. The new approach takes into account both spatial and temporal components at same time in detecting spatio-temporal outliers. Each component is weighted by a parameter that determines how the spatial distance weights and how the temporal distance weights letting each point to be uniquely weighted. In a more precise way, a parameter α defined by the user in the interval $[0, 1]$ is used to determine the influence of the spatial component in the final weight; consequently, $\beta = 1 - \alpha$ is the influence of temporal component. So, this approach allows to work with different kinds of datasets (both those in which the temporal aspects are more relevant and, on the contrary, those in which the spatial one are more important) providing the possibility of managing the weights in an articulated way. The eventual knowledge of the data to be processed will be thus better used.

As said above, the aim is to assign one weight as the linear combination of the spatial weight and the temporal weight.

Firstly the vectors are normalized to obtain data (spatial coordinates and temporal component) in $[0, 1]$ (normalized spatio-temporal representation).

The second step consists of computing a spatio-temporal weight as a weighted linear combination of normalized spatial and temporal weights.

Consider $\omega_{s,k}(q, D)$ the normalized spatial weight of an object q in D computed as the sum of spatial distances, $dist_s$, from its k-nearest *spatial* neighbors $nn_{s,i}(q, D)$, where

the subscript s indicates spatial dependance and k indicates the numbers of neighbors. Normalized temporal weight given to an object q in D is the sum of temporal distances, $dist_t$, from the k-nearest *temporal* neighbors $nn_{t,i}(q, D)$, indicated by $\omega_{t,k}(q, D)$, where the subscript t stands for temporal and k stands for user input parameter k dependance. Hence, for each object q, a spatio-temporal weight is assigned as follows

$$\omega_{s,t,k}(q, D) = \alpha \cdot \omega_{s,k}(q, D) + \beta \cdot \omega_{t,k}(q, D) \tag{4.4}$$

where

$$\omega_{s,k}(q, D) = \sum_{i}^{k} dist_s(q, nn_{s,i}(q, D)) \quad \forall q \in D \tag{4.5}$$

and

$$\omega_{t,k}(q, D) = \sum_{i}^{k} dist_t(q, nn_{t,i}(q, D)) \quad \forall q \in D \tag{4.6}$$

having $\alpha + \beta = 1$.

We would remark that limit cases are

- *Spatial Outlier Detection*
  $\alpha = 1$ and $\beta = 0 \Rightarrow \omega_{s,t,k}(q, D) = \omega_{s,k}(q, D) \ \ \forall q \in D$

- *Temporal Outlier Detection*
  $\alpha = 0$ and $\beta = 1 \Rightarrow \omega_{s,t,k}(q, D) = \omega_{t,k}(q, D) \ \ \forall q \in D$

ST–Outlier Detector Algorithm

1: *begin ST-OutlierDetector*$(D, dist_s, dist_t, n, k, \alpha)$
2: $OutlierStack = null$
3: $WorkingSet = ExtractElements(D)$
4: **while** $(WorkingSet! = null)$ **do**
5: $D = D - WorkingSet$
6: **for** $p \in D$ **do**
7: **for** $q \in WorkingSet$ **do**
8: **if** $(OutlierStack == null\ or\ w_{s,t,k}(q) \geq LowerWeight(OutlierStack))$ **then**
9: $d_s(p,q) = CalculateSpDistance(p, q, dist_s)$
10: $d_t(p,q) = CalculateTempDistance(p, q, dist_t)$
11: $BuildTreeKNN(p, q, d_s, d_t, k)$
12: **end if**
13: **end for**
14: **end for**
15: **for** $q \in WorkingSet$ **do**
16: $w_{s,k}(q) = CalculateWeight(q)$
17: $w_{t,k}(q) = CalculateWeight(q)$
18: $w_{s,t,k}(q) = CalculateCombinedWeight(w_{s,k}(q), w_{t,k}(q), \alpha)$
19: $PushMaxWeights(OutlierStack, w_{s,t,k}(q))$
20: **end for**
21: $WorkingSet = ExtractElements(D)$
22: **end while**
23: **return** $OutlierStack$
24: *end* $STOutlierDetector()$

The algorithm ST–Outlier Detector receives in input the dataset D, containing N objects, the distances $dist_s, dist_t$, the number k of neighbors to consider for the weight computation, the number n of outliers to find, the described parameter α. The algorithm computes the weights of the dataset objects by comparing each object with a small subset of the overall dataset, called Working Set, and storing, for each object, its k-nearest neighbors found in the Working Set. At each step, the weight of an object is thus an upper bound to its true weight because it is the real weight only among the ob-

jects belonging to the Working Set. The objects having a weight lower than the smallest among the n greater weights so far computed will not be considered in future because this condition is sufficient to classify these objects as inliers. At each step, Working Set contains some objects randomly selected from D, among the points of the dataset not processed yet. As the algorithm processes new objects, more accurate weights are computed, because more objects have been taken into account. The algorithm stops when there are no more objects to be processed.
The pseudocode of the *ST–OutlierDetector* algorithm is shown in the following 24. The *BuildTreeK–NN* function stores, for each object p of the dataset D, a structure containing the associated spatial $k\text{-}NNs$ and the temporal $k\text{-}NNs$. The *ExtractElements* function randomly selects a group of objects that will be the next items to be processed. At each iteration, the number of selected objects is a dataset cardinality percentage (about the 4-5 %) experimentally determined. The *ComputeSpDistance* and *ComputeTempDistance* functions compute the spatial and temporal distances respectively, as indicated in equations 4.4 and 4.5, by which the algorithm selects the spatial $k\text{-}NNs$ and the temporal $k\text{-}NNs$. The *ComputeCombinedWeight* function computes the combined weight for each object upon the two spatial and temporal weights as indicated in equation 4.6. The *PushMaxWeights* function marks, as outliers, those objects, belonging to the WorkingSet, having the n maximum weights among those computed. The function stores at the first iteration and updates, at each step, in a stack, named OutlierStack, the objects marked as outliers; the *LowerWeight* function computes the minimum weight among the n maximum weights stored in the OutlierStack.

Complexity

The ST–Outlier Detector Algorithm has worst-case time complexity $O(N^2)$ with $N = |D|$, whenever the real weights have been computed for every object of the overall dataset. Practical complexity is $O(N * (N - M))$ with M represents the number of points for which the real weight has not been computed.

4.3 Experimental Results and Discussion

The proposed approach faces the spatio-temporal outlier detection problem from a new perspective, i.e. considering a mixture of spatial and temporal features like a single feature.

The testing phases on both synthetic and real datasets have been provided interesting results that respect the data characteristics. The tests have been executed on many kind of datasets, of various dimensions. Also many other synthetic datasets generated by the data generator [46] have been used in order to achieve a scaling analysis by growing the volumes of the datasets. Moreover, both trajectory and synthetic example datasets that simulate periodic movements, have been used to evaluate the effectiveness of the proposed approach.

At the best of our knowledge, several real spatial datasets are around for experimental purposes, this is not true in the Trajectory Database domain. There is no available real dataset already explored by a domain expert in order to be used as ground truth for benchmarking. Nevertheless, we have exploited on two synthetic datasets and also on one real–world dataset. Precisely:

1. a small synthetic dataset, named Tracking, collected by us in our laboratory in order to manually inject every kinds of outliers, such as only spatial, only temporal and spatio-temporal outliers;

2. a dataset, named Complex9, [103] that is a benchmarking synthetic dataset, widely used in test phases and publicly available at web site http://www.cs.uh.edu/ sujingwa/PKDD05;

3. a real–world dataset, named School Buses, [35] and publicly available at web site http://www.rtreeportal.org.

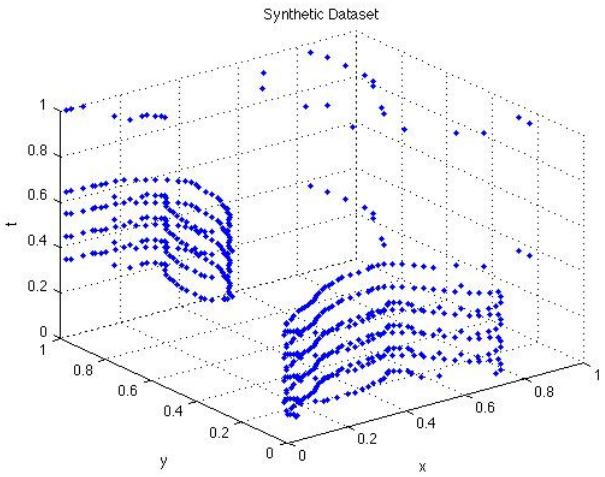
4.3.1 Tracking dataset

The Tracking dataset is a synthetic dataset that simulates same periodic trajectories with some added outliers. Tracking dataset consists of 9 trajectories of 2 moving objects for 2 distinct days. The structure of each pattern is as follows:

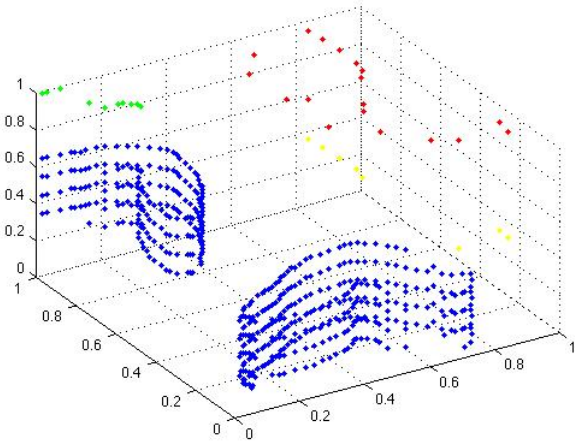
$$\{obj\_id, traj\_id, date, time, x, y\}$$

where obj_id is a numeric identification of the moving object, $traj\_id$ is a unique trajectory identification, (x, y) is the position of moving object in a cartesian reference system, the date is expressed in $dd/mm/yyyy$ format and the time is expressed in hh:mm:ss format. The obj_id and $traj\_id$ are not considered, while the two fields date and time are

converted in an only one field t consisting of the serial date number for the corresponding elements year, month, day, hour, minute and second. Hence, the dataset is shown in



(a)



(b)

Figure 4.1: Tracking dataset: (a) Normalized representation (b) Outliers marked with different colors

figure 4.1(a), in a 3D cartesian reference system, in which x and y are spatial coordinates

Dataset	Entry	Attrib.	ST-O	S-O	T-O	# Inliers	# Outliers
Tracking	602	6	18	30(12+18)	28(10+18)	562	40(18+12+10)

Table 4.1: Tracking dataset: Details

and the third dimension is the time t. In the figure 4.1(b) the objects have been drawn with different colors: inlier data are blue marked points, spatio-temporal outliers are red marked points, only spatial outliers are yellow marked points, only temporal outliers are green marked points.

As shown in table 4.1, the number of added outliers is 40. So, as in our approach, the required outlier number, indicated by n, is an input parameter, we can set it to:

- $n = 30$ in case of spatial outlier detection
- $n = 28$ in case of temporal outlier detection
- $n = 40$ in case of spatio-temporal outlier detection

to keep into account, in this last case, all the outlier objects.

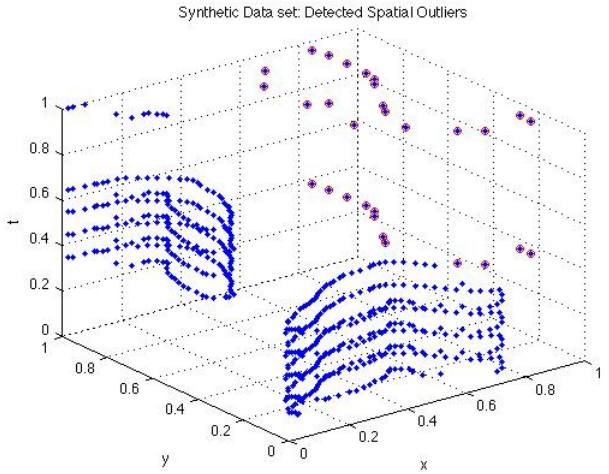
Limit case: Spatial Outlier Detection

Spatial Outlier Detection Parameter Settings

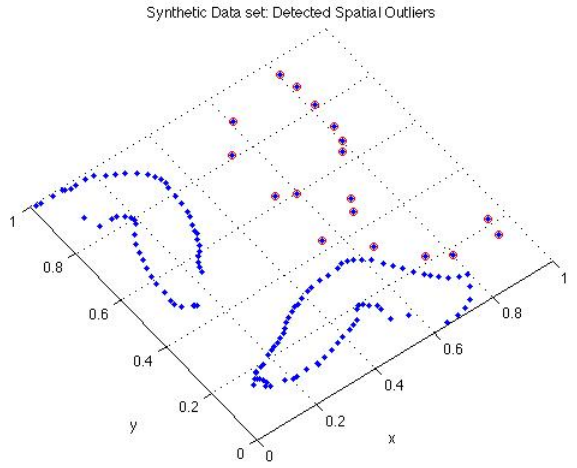
- $OutlierNumber = 30$
- $NearestNeighborNumber = 10$
- $\alpha = 1$
- $\beta = 0$

The detection of 30 objects has been obtained as result in this case. These objects are the more distant (spatially) from the rest of the data, those red circled in figure 4.2(a).

A 2D-plotting of the dataset visualizes better the meaning of the obtained result: indeed, in this test case, only spatial coordinates are involved (figure 4.2(b)).

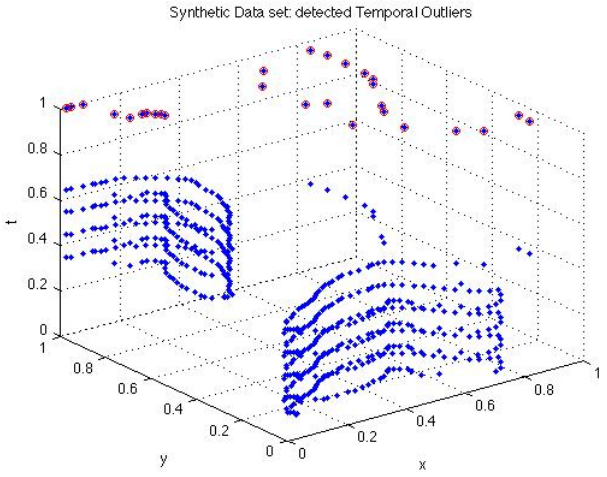


(a)

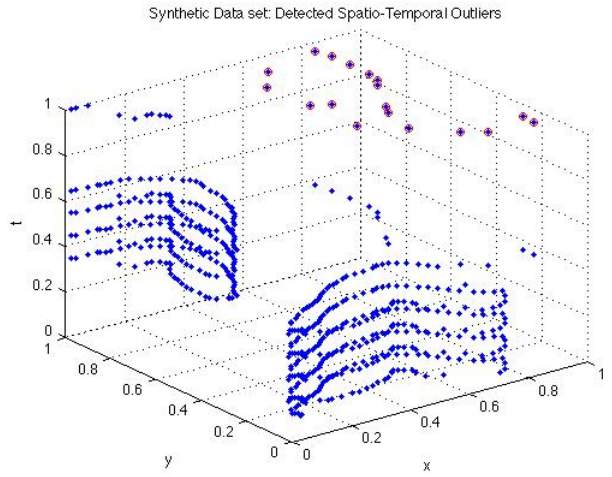


(b)

Figure 4.2: Tracking dataset: (a) Detected Spatial Outliers (b) 2D-plotting



(a)



(b)

Figure 4.3: Tracking dataset: (a) Detected Temporal Outliers (b) Detected Spatio-Temporal Outliers

Limit case: Temporal Outlier Detection

Temporal Outlier Detection Parameter Settings

- $OutlierNumber = 28$
- $NearestNeighborNumber = 10$
- $\alpha = 0$
- $\beta = 1$

The obtained result is very complaisant with dataset analysis reported in the table 4.1 above: the result correctness can be verified from the figure 4.3(a) where the detected temporal outliers have been red circled.

Spatio-Temporal Outlier Detection

Spatio-Temporal Outlier Detection Parameter Settings

- $OutlierNumber = 40$
- $NearestNeighborNumber = 10$
- $\alpha = 0.5$
- $\beta = 0.5$

The obtained result is shown in the figure 4.3(b) where the detected spatio-temporal outliers (the first 18 having the higher weights among the 40 required) have been red circled.

Dataset	Entry	Attributes	ST-O	S-O	T-O	# Inliers	# Outliers
School Buses subset	30000	8	N.A.	800	100	29100	900

Table 4.2: School Buses Subset: Details

4.3.2 School Buses dataset

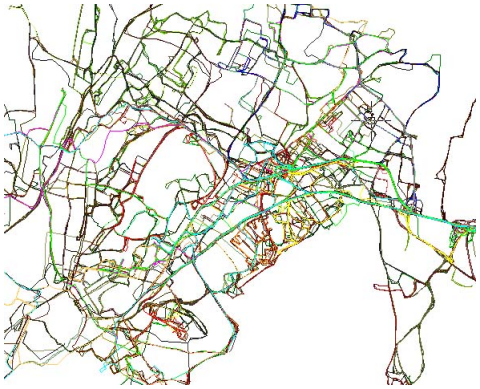
The real-world dataset, named School Buses, consists of 145 trajectories of 2 school buses collecting and delivering students around Athens metropolitan area in Greece for 108 distinct days.
This real dataset are publicly available at web site: http://www.rtreeportal.org and contains about 69000 entries.

The structure of each record is as follows:

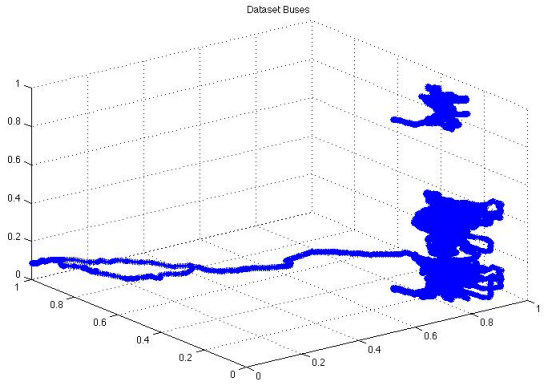
$$\{obj\_id, traj\_id, date, time, lat, lon, x, y\}$$

where obj_id is the school bus identification, $traj\_id$ is the unique trajectory identification, the date and time are the sampling timestamp (every 30 seconds), the date in $dd/mm/yyyy$ format and the time in hh:mm:ss format, the (lat, lon) and (x, y) are the bus location, in WGS84 reference system and in GGRS87 reference system, respectively. The obj_id and $traj\_id$ are not considered, the two fields date and time are converted in an only one field t consisting of the serial date number for the corresponding elements year, month, day, hour, minute and second. Moreover, the lat and lon are redundant and so are not considered too; indeed x and y give the same information. Hence, the normalized representation of the dataset is illustrated in figure 4.4(b): in a 3D cartesian reference system, x and y are the spatial coordinates and the third dimension is the time t. In figure 4.4(a) the trajectory map of School Buses is shown.

The outlier detection is devoted to spatial and temporal outlier detection, because each point has a unique weight depending on both components. If the parameter α grows hence the goal is to give major weight to spatial outliers, if the parameter α is kept lower (and consequently the parameter β grows), hence the goal is to give major



(a)



(b)

Figure 4.4: School Buses dataset: (a) Map (b) Normalized representation

weight to temporal outliers.

As said before, this real dataset is not analyzed and labeled by a domain expert; hence, in order to understand better the obtained results, we have selected a data subset from the original dataset, consisting of about 30000 entries, and a dataset analysis, particularly about temporal feature, in order to understand better the obtained results, has been conducted and provided in the appendix. Some only temporal outliers have been also injected. The data subset, to be used during the tests, is shown in the following figure 4.5,

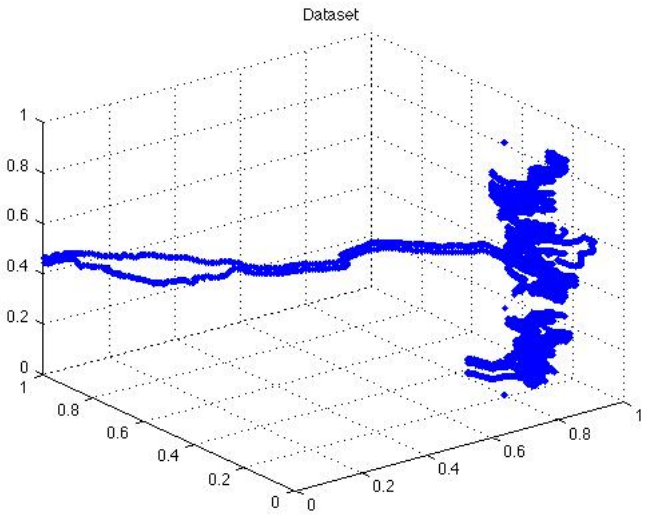


Figure 4.5: School Buses Subset: a subset with added temporal outliers

and is also reported in table 4.2.
In the following, significative test cases have been executed in order to detect Spatial and/or Temporal Outliers. The required outlier number, indicated by n, is an input parameter, so it will be set to:

- $n = 800$ in case of spatial outlier detection
- $n = 100$ in case of temporal outlier detection
- $n = 800$ in case of spatio-temporal outlier detection.

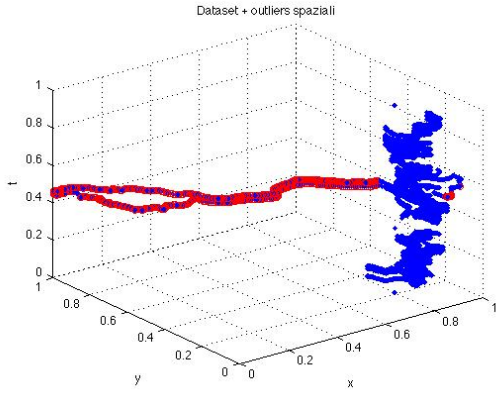
Limit case: Spatial Outlier Detection

Spatial Outlier Detection Parameter Settings

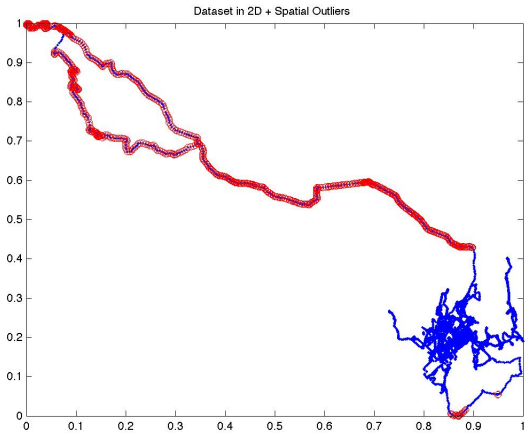
- $OutlierNumber = 800$
- $NearestNeighborNumber = 300$
- $\alpha = 1$

- $\beta = 0$

The expected result of this test case is 800 objects detection - spatial outliers - the more distant (spatially) from the distribution.



(a)



(b)

Figure 4.6: School Buses subset: Detected Spatial Outliers (a) 3D plotting (b) 2D plotting

The detection of 800 objects, those red circled in figure 4.6(a), has been obtained as result. A 2D-plotting of the dataset visualizes better the meaning of the obtained result:

indeed, in this test case, only spatial coordinates are involved. The objects, drawn in red, are really the spatially farther than the rest of the data as shown in figure 4.6(b).

Limit case: Temporal Outlier Detection

Temporal Outlier Detection consists of object detection that have not enough temporal neighbors. The neighbor number is an input parameter, that has been set by a very low value in order to work like a filter, dropping out isolated objects.

Temporal Outlier Detection Parameter Settings

- $OutlierNumber = 100$
- $NearestNeighborNumber = 300$
- $\alpha = 0$
- $\beta = 1$

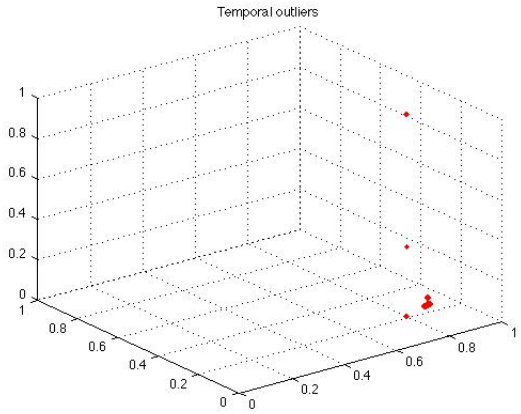
The obtained result is very complaisant with dataset analysis reported in the table below: as the first 100 top outliers, the added outliers (12 objects in total) plus 88 objects belonging to sample groups of days 24/10 and 26/10 will be detected. The result correctness can be verified both from table and from the following figure 4.7(a). Indeed, the detected dates have no enough (less than 300) neighbors, respectively 78 and 94.

Temporal outliers are the red circled among the dataset, as shown in figure 4.7(b).

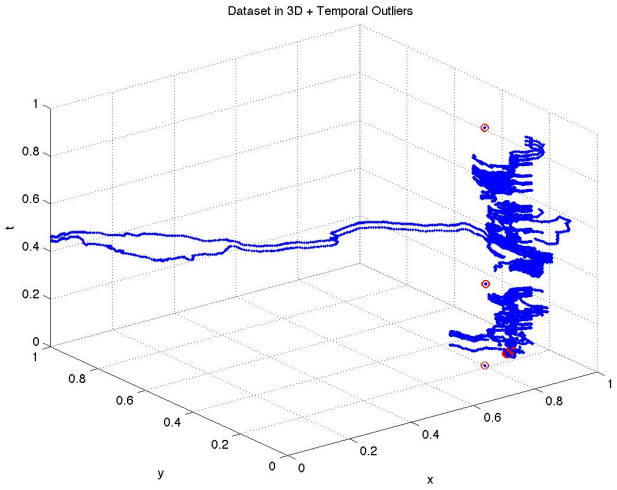
General case: Spatio-Temporal Outlier Detection

This test case is the combined case with both parameters α e β different from zero.
Spatio-Temporal Outlier Detection Parameter Settings

- $OutlierNumber = 800$
- $NearestNeighborNumber = 300$
- $\alpha = 0.5$
- $\beta = 0.5$



(a)



(b)

Figure 4.7: School Buses subset: (a) Temporal Spatial Outliers (b) The subset with temporal outliers

The result will be shown in figure 4.8. We would remark that the required outlier number is still 800 such as the spatial case, so we do not keep into account all outliers detected in only spatial case and in only temporal one. The right outlier number should be 900.

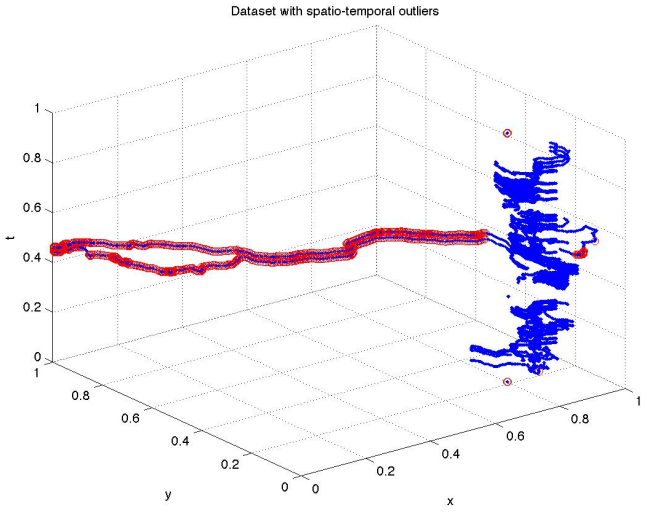


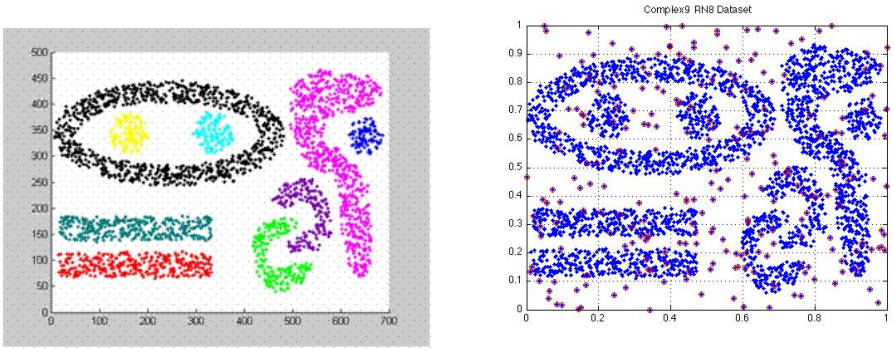
Figure 4.8: School Buses subset with detected spatio-temporal outliers

So, as expected, we loose same objects from temporal outliers and some from spatial ones, detecting the more relevant of both.
Other test cases have been executed: we report some considerations about obtained results.
Keeping fixed the other parameters (outlier and nearest neighbor number), the parameter α has been set to many values, so we can distinguish some two ranges: the former is $[1, 0.8]$ in which the result is almost similar to only spatial case, the latter is $[0, 0.2]$ in which the result is almost similar to only temporal case; on the contrary, in the middle interval $[0.2, 0.8]$, it is possible to taste the effect of mixed weights.

4.3.3 Complex9 dataset

The dataset, called Complex9 [103], that is a benchmarking synthetic dataset widely used in test phases and publicily available. Complex9 has two attribute values and nine classes, as shown in figure 4.9(a). Six different versions of the Complex9 dataset are available by adding noise examples with different size and type. The first three, Complex9_RN8, Complex9_RN16, and Complex9_RN32, were created by adding $8\%, 16\%, 32\%$ random noise

examples to Complex9 dataset. The second three, Complex9_GN8, Complex9_GN16, and Complex9_GN32, were created by adding $8\%, 16\%, 32\%$ gaussian noise examples to Complex9 dataset. The Complex9_RN8 dataset is a spatial dataset that has been used



(a) (b)

Figure 4.9: (a) Normal Complex9 dataset version (b) Normalized noise version: Complex9_RN8.

to test spatial outlier detection (a limit case). Then, we added the temporal component in order to obtain a synthetic spatio-temporal dataset called Complex9_RN8_time, shown in figure 4.10 that is used to test temporal outlier detection (other limit case) and spatio-temporal outlier detection.

The test phase foresees also the comparison, in terms of accuracy, with two other techniques:

DBScan, [32], a clustering density based technique resistant to noise, on which is based a spatio temporal outlier detection technique ST-DBScan [21] and LDBOD (Local distribution based outlier detector) [111] a very recent outlier detection technique chosen among the others proposed in literature because based on LOF (Local Outlier Factor) [114] and *k-nearest neighbors* techniques.

As shown in table 4.3, the number of added outliers is 242, so, as in our approach, the required outlier number is an input parameter, we set it to: 242 first, then to 200 and to 136. In the table 4.4 the obtained results have been shown. The results have shown an high accuracy, that is better as the outlier number required is going down. Indeed, the random noise injected sometimes is not real noise due to its random nature; so the

Dataset	Entry	Attribute	Class	Added Outliers
Complex9	3031	2	9	0
Complex9_RN8	3273	2	9	242 (spatial outliers)
Complex9_RN8_Time	3273	3	9	26 (temporal outliers)

Table 4.3: 2D and 3D-dataset used: Details

effective noisy data are less than 242, as we can observe in figure 4.9(b), some red circles fall inside the inlier data. Now, we want to compare with DBScan and LDBOD.

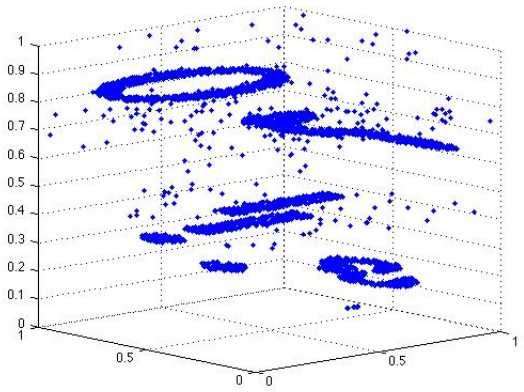


Figure 4.10: Complex9_RN8_Time dataset

Unlike ST-Outlier Detector and LDBOD, DBScan doesn't require as input parameter the number of outlier, so we execute DBScan on Complex9_RN8 dataset and tuning its input parameters ($minPts = 11 \quad and \quad \varepsilon = 0.025$) in order to obtain 9 clusters (as nine

Dataset	**K-NN/ MinPts**	**Required Outliers/ Obtained Outliers**	**ST-Outlier Detector**	**DBScan**
Complex9_RN8	11	242	0.941320	N.A.
Complex9_RN8	11	200	0.951711	N.A.
Complex9_RN8	11	136	0.959658	0.953545

Table 4.4: Spatial Outlier Detection: Classification Accuracy of ST-Outlier Detector and DBScan

Dataset Name	K-NN/ MinPts	Outlier Req./Obt.	ST-Outlier Detector	DBScan
Complex9_RN8_Time	11	26	1.000000	1.000000

Table 4.5: Temporal Outlier Detection: Classification Accuracy of ST-Outlier Detection and DBScan

classes), the number of obtained outliers is 136. So, now, we set as required outlier number (input parameters for ST-Outlier Detector and LDBOD) to 136 and we can compare the accuracy. As shown in the last row, the accuracy of ST-Outlier Detector is higher than DBScan.
In the following cases, Complex9_RN8_Time dataset is used in which temporal component and 26 temporal outliers have been added.
Similarly, in table 4.5 we list the obtained results, in terms of accuracy for Temporal Outlier Detection for ST-Outlier Detector and DBScan. Both methods report the same maximum accuracy.
In table 4.6, we list the obtained results, in terms of accuracy for Spatio-Temporal Outlier Detection, also for LDBOD. The true outlier number is 17, so we compare ST-Outlier Detector and LDBOD setting as spatio–temporal outlier number, input parameter, to 17. As shown in the table, ST-Outlier Detector reports a better accuracy than LDBOD.
In order to compare these two methods also with DBScan,we made another running of ST-Outlier Detector and LDBOD setting as spatio–temporal outlier number (input parameter) to 243. ST-Outlier Detector reports a little bit better accuracy respect to DBScan and LDBOD, and DBScan reports a better accuracy respect to LDBOD.

Dataset name	K-NN/ MinPts	Out. Req./Obt.	ST-Outlier Detector	DBScan	LDBOD
Complex9_RN8_Time	11	17	0.991443	N.A.	0.990220
Complex9_RN8_Time	11	243	0.931235	0.930929	0.929401

Table 4.6: Spatio-Temporal Outlier Detection: Classification Accuracy of ST-Outlier Detection, DBScan and LDBOD

4.4 Outlierness Degree Mapping

As previously described, the input parameter determination can be a non trivial task, but the quality of the results depends on an appropriate choice of the parameters. So, in this section, another contribution, called the **outlierness degree map**, has been proposed. This is a visualization tool that allows to make a 3D-plot of the points belonging to dataset by drawing them with different colors and also different color nuance based upon their outlierness degree. The map is built without setting, a-priori, outlier number to be found. The main goal of this kind of analysis is to visualize the dataset structure with respect to outlier presence. Starting from blue color, used for the inliers, i.e. the object with minimum outlierness degree, passing through cyan, green, yellow, until red color that represents the outliers, i.e. the objects with maximum outlierness degree. Also the same color can have different nuances with the same meaning. Based upon α and *Nearest Neighbor Number* parameter values, different mappings have been obtained. The same test cases, such as the experimental tests proposed for the school buses dataset in section 4.3.2, are proposed again here.

Spatial Outlierness Degree Mapping

Configuration settings for spatial outlierness degree map

- $Nearest\ Neighbor\ Number = 300$
- $\alpha = 1$
- $\beta = 0$

The result has been visualized in the following figure 4.11.

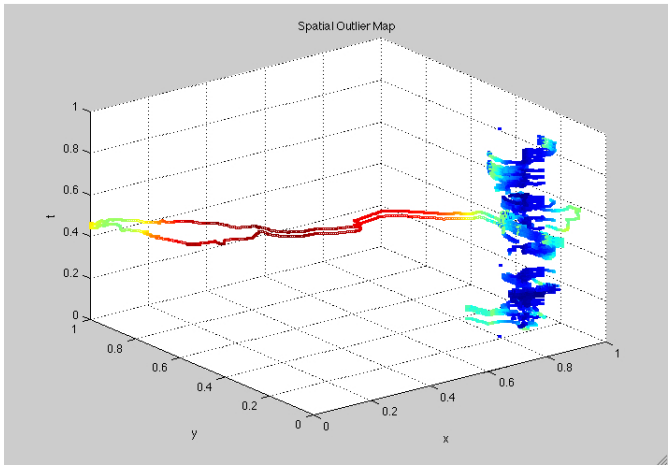


Figure 4.11: School Buses dataset: Spatial Outlierness Mapping

The dark red, light red, yellow and light green colors represent the two trajectories that are spatially abnormal respect to the rest of data. The light blue and light green colors represent the trajectory queues, the dark blue color is used for the central data (spatial inliers) and also for added temporal outliers, because in this case time is not relevant. The blue color became darker going closer to the center of the distribution.

Temporal Outlierness Degree Mapping

Configuration settings for temporal outlierness degree map

- $Nearest\ Neighbor\ Number = 100$

- $\alpha = 0$
- $\beta = 1$

The result has been visualized in the following figure 4.12.

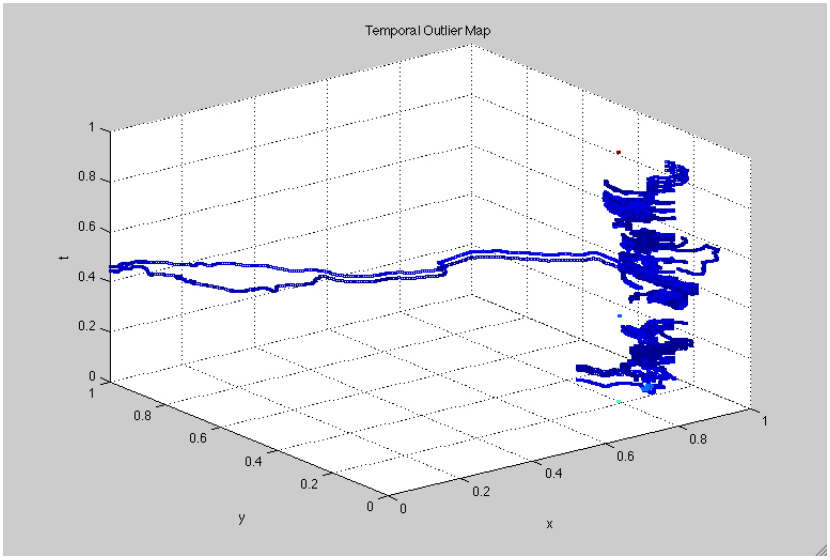


Figure 4.12: School Buses dataset: Temporal Outlierness Mapping

The red, light green and cyan colors represent the three added temporal outlier groups. The cyan color also represents the other objects that have less than 100 neighbors; the rest of dataset are drawn by blue color (also only spatial outliers).

Spatio-Temporal Outlierness Degree Mapping

Case 1

Configuration settings for spatio-temporal outlierness degree map

- $Nearest\ Neighbor\ Number = 300$
- $\alpha = 0.5$

- $\beta = 0.5$

The result has been visualized in the following figure 4.13.

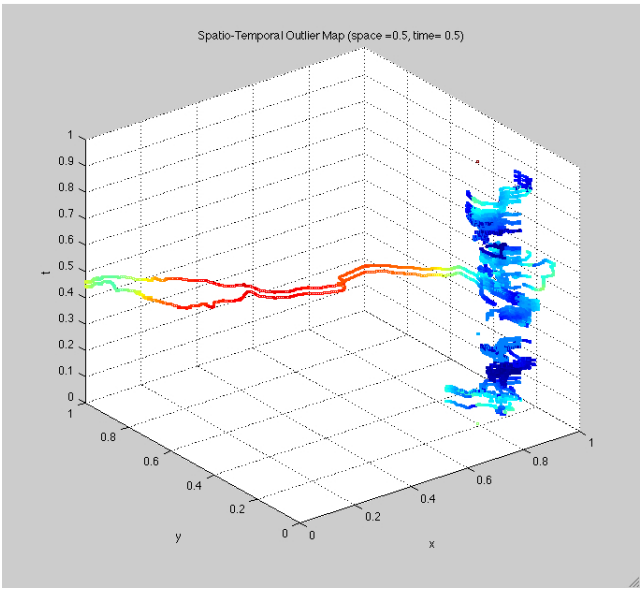


Figure 4.13: School Buses dataset: Spatio-Temporal Outlierness Mapping $\alpha = 0.5$

The light red, light green and yellow colors represent the two trajectories that are spatially abnormal respect to the rest of data and also the three added temporal outlier groups. The dark red is used for the top group of added temporal outliers. The cyan color also represents the other objects that have less than 100 neighbors; the rest of dataset are drawn by blue color.

Case 2

Configuration settings for spatio-temporal outlierness degree map

- $Nearest\ Neighbor\ Number = 300$
- $\alpha = 0.8$
- $\beta = 0.2$

The result has been visualized in the following figure 4.14.

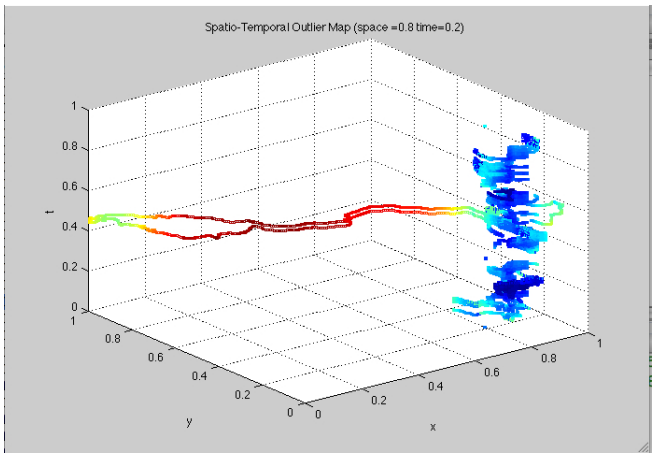


Figure 4.14: School Buses dataset: Spatio-Temporal Outlierness Mapping $\alpha = 0.8$

This test case ($\alpha = 0.8$) is similar to only spatial outliers detection with $\alpha = 1$. The red color, light green and yellow colors represent the two trajectories that are spatially abnormal respect to the rest of data while the three added temporal outlier groups are drawn by cyan or light blue colors, they have a small degree of outlierness.
If we compare this result with the previous one, we can observe that giving a major importance to spatial weight, the color nuances change: the light red became dark red, and light blue became dark blue changing from $\alpha = 0.5$ to $\alpha = 0.8$.

4.5 Summary

A novel *non parametric approach* to face the outlier detection problem in an unlabeled spatio-temporal dataset has been presented. It combines spatial and temporal attributes in order to find out a user-defined number of top spatio-temporal outliers.
The method has been proved on synthetic and real datasets to be efficient in space and time to detect the spatio-temporal outliers.
The strenght of this approach is to combine two different features, depending on different aspects, without fixing any spatial and/or temporal criterion. As shown above, the weakness is about the choice of input parameter α in the general case and the number of outliers required. To partially address to this problem, the development of a framework in which, without fixing the number of outliers required, we are able to plot the entire description map of the analyzed dataset have been illustrated. The map is a visual analytics tool that provide a user supporting tool to better choose the number of outliers required.

Examples of application fields of this work are:

- a filter to pre-process with respect to spatial and/or temporal features high dimensionality datasets in order to eliminate inconsistent ST-data.
- an anomaly detection process for objects belonging to trajectory datasets or other kind of spatio-temporal datasets.

5 A Rough Set Approach to ST-Outlier Detection

5.1 Introduction

Spatio-temporal data mining is a growing research area dedicated to the development of algorithms and computational techniques for the analysis of large spatio-temporal databases and the disclosure of interesting and hidden knowledge in these data, mainly in terms of periodic hidden patterns and outlier detection.
In the meantime, an emerging conceptual and computing paradigm of information processing, granular computing has received much attention recently. Many models and methods of granular computing have been proposed and studied. In this chapter, we are going to introduce a new approach to spatio-temporal outlier detection, which exploits granular computing potentialities. There are several types of granularity encountered in data mining and machine learning: from a concept granulation point of view, the origins of the granular computing ideology are to be found in the rough set literature. Hence, the attention of this novel approach has been focalized on outlier detection in spatio-temporal data using rough set theory.
Rough set theory introduced by Z. Pawlak, as an extension of naive set theory, is for the study of intelligent systems characterized by insufficient and incomplete information. It is motivated by practical needs in classification and concept formation. In recent years, there has been a fast growing interest in rough set theory.

Unlike most current methods for outlier detection exploit rough set theory to define new rough weights as degree of outlierness, as described in the section 3.4. our goal is

to represent the Outlier Set such as a rough set through its lower, upper approximation, remarking the benefits of keeping into account the objects belonging to the boundary. Moreover, we introduce a new set, called Kernel Set. This set is a selected subset of elements that is able to describe the original dataset both in terms of data structure and in terms of obtained results. In particular, we want to show the advantages of considering the Kernel Set. Indeed, we compare the Rough Outlier Set extracted by the entire Data Set and the Rough Outlier Set extracted by the simple Kernel Set.
With this aim, this chapter is organized as follows.
In the next section, some preliminaries about rough set theory that are relevant to our approach are reported.
Then, the new rough set approach, called ROSE (Rough Outlier Set Extraction), to detect spatio-temporal rough outlier set is described.
Executed tests on real world datasets and performance evaluation of the algorithm are shown and finally, conclusion remarks are given in the last section about ongoing and future work.

5.2 Rough Set Theory

In this section some basic elements of Rough set Theory, relevant for our scope.

5.2.1 Indiscernibility and Set Approximation

Let U the universe of the discourse and A the finite and non empty set of attributes, then $S = \langle U, A \rangle$ is an information system.
Let B a proper subset of A. With every subset of attributes $B \subseteq A$, one can easily associate an equivalence relation I_B on U:

$$I_B = \{(p, q) \in U \times U \ / \ \forall a \in B, \quad a(p) = a(q)\} \tag{5.1}$$

I_B is called *B-indiscernibility relation.*
If $(p, q) \in I_B$, then objects p and q are indiscernible from each other by attributes B. The equivalence classes of the partition induced by the *B-indiscernibility relation* are denoted by $[p]_B$. These are also known as *granules.*

We can approximate any subset X of U using only the information contained in B by constructing the lower and upper approximations of X.
The sets $\{p \in U : [p]_B \subseteq X\}$ and $\{p \in U : [p]_B \cap X \neq \emptyset\}$, where $[p]_B$ denotes the equivalence class of the object $p \in U$ relative to I_B , are called the *B-lower* and *B-upper approximation* of X in S as shown in Figure 5.1, taken by [63], and respectively denoted by $\underline{B}(X), \overline{B}(X)$. The objects in $\underline{B}(X)$ can be certainly classified as members of X on the basis of knowledge in B, while objects in $\overline{B}(X)$ can only be classified as possible members of X on the basis of B.

5.2.2 Dependency Rule Generation

Reducts

Indiscernibility relation reduces the data by identifying equivalence classes (objects that are indiscernible), using the available attributes. Only one element of the equivalence

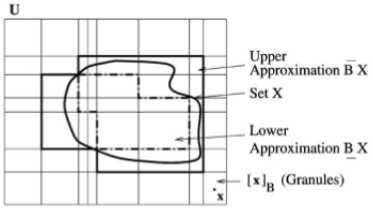


Figure 5.1: Lower and Upper Approximation

class is needed to represent the entire class. Reduction can also be done by keeping only those attributes that preserve the indiscernibility relation and, consequently, set approximation. The above sets of attributes are called *reducts* as reported in [63]. Reducts have been characterized by discernibility matrices and discernibility functions.
Let us consider an information system $S =< U, A >$ with $U = \{p_1, ..., p_n\}$ our universe of the discourse and $A = \{a_1, ..., a_m\}$ the set of attributes. Then, p_i is a m-dimensional feature vector and $a_j, j = 1, ..., m$ the attributes.
The discernibility matrix $M(S)$ of S is meant as an $n \times n$ (symmetrical with empty diagonal) matrix with entries c_{ij}:

$$c_{ij} = \{a \in A \ / \ a(p_i) \neq a(p_j)\}$$

A discernibility function f_S is a function of m Boolean variables $\overline{a}_1, ..., \overline{a}_m$ corresponding to the attributes $\{a_1, ..., a_m\}$, and defined as follows:

$$f_S(\overline{a}_1, ..., \overline{a}_m) = \wedge\{\vee(c_{ij}) : 1 \leq i, j \leq n, j < i, c_{ij} \neq \emptyset\} \tag{5.2}$$

where $\vee c_{ij}$ is the disjunction of all variables $\overline{a}$ with $a \in c_{ij}$. It is seen that $\{a_{i1}, ..., a_{ip}\}$ is a *reduct* in S if and only if $a_{i1} \wedge ... \wedge a_{ip}$ is a prime implicant (constituent of the disjunctive normal form) of f_S.

Methodology

A principal task in the method of rule generation is to compute *reducts* and *discernibility matrix* relative to a particular kind of information system, i.e. the *decision system*,

getting, respectively, names *d–reducts* and *d–discernibility matrix*. The methodology is described below.

A *decision system* is an *information system* $S =< U, A >$ with $A = C \cup \{d\}$, where C and $\{d\}$ are respectively sets of condition attributes and of decision.

Let the value set of d be of cardinality l, i.e.,

$$V_d = \{d_1, d_2, ..., d_l\},$$

where l represents the number of classes. The *decision system* $S =< U, A >$ may be divided into l tables:

$$S_i =< U_i, A_i >,\ i = 1, ..., l,$$

corresponding to the l decision attributes $d_1, ..., d_l$, where $U = U_1 \cup ... \cup U_l$ and $A_i = C \cup \{d_i\}$.

Let $\{p_{i_1}, ..., p_{i_p}\}$ be the set of those objects of U_i that occur in $S_i, i = 1, ..., l$.

Now, for each d_i-*reduct* $B = \{b_1, ..., b_k\}$, a d_i-*discernibility matrix*, denoted by $M_{d_i}(B)$, can be derived as follows:

$$c_{ij} = \{a \in B : a(p_i) \neq a(p_j)\},\ \forall i, j = 1, ..., n.$$

For each object $p_j \in p_{i_1}, ..., p_{i_p}$, the discernibility function $f_{d_i}^{p_j}$ is defined as

$$f_{d_i}^{p_j} = \wedge\{\vee(c_{ij}) : 1 \leq i, j \leq n, j \leq i, c_{ij} \neq \emptyset\}$$

where $\vee c_{ij}$ is the disjunction of all members of c_{ij}. Then, $f_{d_i}^{p_j}$ is brought to its disjunctive normal form (d.n.f).

A dependency rule $r_i : d_i \leftarrow P_i$ has been obtained, where P_i is the disjunctive normal form (d.n.f) of $f_{d_i}^{p_j}$, $j \in i_1, ..., i_p$.

Let B_i be the set of condition attributes occurring in the rule r_i, so the dependency factor df_i for r_i is defined as:

$$df_i = \frac{card(POS_{B_i}(d_i))}{card(U_i)} \tag{5.3}$$

where $POS_{B_i}(d_i)$ is the positive region of class d_i with respect to attributes B_i, given by: $POS_{B_i}(d_i) = \bigcup_{X \in I_{d_i}} \underline{B_i}(X)$, where $\underline{B_i}(X)$ is the lower approximation of X with respect to B_i.

The dependency factor df_i gets values in the interval $[0, 1]$, with the maximum and minimum values corresponding, respectively, to complete dependence and independence of d_i on B_i.

5.3 Spatio-Temporal Outlier Detection Problem

5.3.1 Theory

Let us consider an information system $S =< U, A >$ with U a spatio-temporal normalized dataset and A its set of attributes. U can be written as follows:

$$U = \{p_i \equiv (z_{i1}, z_{i2}, ..., z_{im}) \in [0,1]^m, \quad i = 1, ..., N\}$$

where $p_i,\ i = 1, ..., N$ is a m-dimensional feature vector.
$A = \{a_1, a_2, a_3, ..., a_m\}$ is the attribute set: a_1, a_2 are the spatial attributes, a_3 is the temporal one and $a_4, ..., a_m$ are other attributes. Hence, a spatio-temporal dataset possesses, at least, three attributes as follows:

$$U = \{p_i \equiv (z_{i1}, z_{i2}, z_{i3}) \in [0,1]^3, \quad i = 1, ..., N\}$$

where $p_i,\ i = 1, ..., N$ is a 3-dimensional feature vector and $A = \{a_1, a_2, a_3\}$ is the minimum attribute set.
Given U, an integer $n > 0$ and a measure $d_{p_i}(U)$, over every $p_i \in U$, the formal definition of the **Outlier Detection Problem** is the following:

Definition 8. *The Outlier Detection Problem consists of finding the* $\overline{n} \geq n$ *objects*

$p_1, p_2, ..., p_n, p_{n+1}, ..., p_{\overline{n}} \in U$ *such that*

$$d_{p_1}(U) \geq d_{p_2}(U) \geq ... \geq d_{p_n}(U) = d_{p_{n+1}}(U)... = d_{p_{\overline{n}}}(U) > d_{p_j}(U) \quad \forall j = \overline{n}+1, ..., N$$

According to this definition, the concept of measure is used to determine the degree of dissimilarity of each object with respect to all others. Then, the *n-Outlier Set* can be formally defined as:

Definition 9. *A n-Outlier Set* $O \subseteq U$ *is the set of* $\overline{n} \geq n$ *objects:*

$$O = \{p_1, ..., p_n, p_{n+1}, ..., p_{\overline{n}} \in U : d_{p_1}(U) \geq ... \geq d_{p_n(U)} = d_{p_{n+1}}(U).. = d_{p_{\overline{n}}}(U) >$$

$$d_{p_j}(U) \quad \forall j = \overline{n}+1, ..., N\}$$

where $d_{p_i}(U),\ \forall i = 1, ..., N$ *is a measure defined and computed on* U.

From the definition 9 it follows that

$$\tau = d_{p_n}(U) \tag{5.4}$$

is the **outlierness threshold**, i.e. the minimum value among the n maximum values of measures computed in U (associated to objects belonging to the *n-Outlier Set*) i.e.

$$\tau = min\{max_1(d_p(U), d_q(U)), ..., max_n(d_p(U), d_q(U))\}\ \forall\ p, q \in U \tag{5.5}$$

Starting from the definition of spatial outlier and temporal outlier due to Birant and Alp [21] that says:
"a spatial outlier is a spatial referenced object whose non-spatial attribute values are significantly different from those of other spatially referenced objects in its spatial neighborhood", and
"a temporal outlier is an object whose non-spatial attribute value is significantly different from those of other objects in its temporal neighborhood", we propose the following definitions applied only to spatio-temporal data:

Definition 10. *A Spatial Outlier (**S-Outlier**) is an object whose spatial attribute value is significantly different from those of its closer objects (spatial neighborhood).*

In this framework, the *Spatial Outlier* definition corresponds to:

Definition 11. *Given* U, *an integer* $n > 0$ *and a measure on spatial component* $d^s_{p_i}(U)$, *defined over every* $p_i \in U$, *an object* $p \in U$ *is a **S-Outlier*** $iff\ \ d^s_p(U) \geq \tau$ *where* τ *is defined in (5.5).*

Following definition 11, it holds that:

Proposition 1. *A Spatial Outlier (**S-Outlier**) is an object that belongs to the spatial n-Outlier Set indicated by* O_s.

Similarly, we propose the following definition of *Temporal Outlier*, applied to only spatio-temporal data:

Definition 12. *A Temporal Outlier (**T-Outlier**) is an object whose temporal attribute value is significantly different from those of its closer objects (temporal neighborhood).*

In this framework, the *Temporal Outlier* definition corresponds to:

Definition 13. *Given U, an integer $n > 0$ and a measure on temporal component $d^t_{p_i}(U)$, defined over every $p_i \in U$ defined on U, an object $p \in U$ is a **T-Outlier** iff $d^t_p(U) \geq \tau$ where τ is defined in (5.5).*

Equality, following definition 13 it holds that:

Proposition 2. *A Temporal Outlier (**T-Outlier**) is an object that belongs to the temporal n-Outlier Set indicated by O_t.*

Definition 10 states that a spatial outlier has no objects or a small group of objects in its spatial neighborhood.
The same is valid for a temporal outlier according to Definition 12. Following both definitions the following holds:

Definition 14. *A Spatio-Temporal Outlier (**ST-Outlier**) is an object which respects both the definitions above.*

To obtain a real degree of outlierness, an appropriate measure should be associated to each object; i.e. the Euclidean distance computed between each object and all the other objects belonging to U. In real applications, with huge amount of data, this idea is unfeasible due to its high computational complexity ($O(N^2)$) where $N = |U|$.
In this approach, we preserve two aims: on one hand, we exploit the well-known outlier

definition based on k-nearest neighbors [70], in order to associate to each object, a measure based on the distances among the object itself and its k-nearest neighbors rather than from all N objects with $k \ll N$. On the other hand, we make use of a pruning strategy that discards objects that surely cannot belong to the *n-Outlier Set*, in order to address the problem of alleviating the computational cost.
In a Spatio–Temporal context, the measure associated to each object is based upon the distances from its spatial k-nearest neighbors and its temporal k-nearest neighbors [69]. Precisely:

$$d_p^{s,t}(U) = \alpha \cdot d_p^s(U) + \beta \cdot d_p^t(U) \tag{5.6}$$

where:

$$d_p^s(U) = \sum_{j=1}^{k} d^s(p, N^s(p, p_j)), \quad \forall p \in U \tag{5.7}$$

$$d_p^t(U) = \sum_{j=1}^{k} d^t(p, N^t(p, p_j)), \quad \forall p \in U \tag{5.8}$$

$k > 0$ is the number of nearest neighbors to keep into account, $N^s(p, p_j)$ and $N^t(p, p_j)$ are, respectively, the *j-th* spatial nearest neighbor and the *j-th* temporal nearest neighbor of p, and α, β weight such that $\alpha + \beta = 1$. The Definition 8, that introduces the Outlier Detection Problem, defines the Spatio-Temporal Outlier Detection Problem, by selecting a measure as in 5.6.
To better illustrate the ideas, the previous and the following definitions, let us introduce a spatio-temporal dataset, called Example and indicated by E:

$$\mathrm{E} = \{p_i \equiv (z_{i1}, z_{i2}, z_{i3}) \in [0,1]^3, \quad i = 1, ..., 18\}$$

where p_i is a 3-dimensional feature vector and $A = \{a_1, a_2, a_3\}$ is the essential attribute set, i.e. a_1, a_2 are the spatial attributes and a_3 is the temporal attribute.
E is a labeled dataset, containing 18 elements, as reported in table 6.2 in the appendix and plotted in the Figure 5.2.

By fixing $k = 3$ and $n = 4$, the outlier sets (spatial, temporal outlier sets), on the basis of the previous definitions, are computed in the examples 1, 2 respectively.

A 4-*Spatial Outlier Set* $O_s \subseteq E$ is the set of objects $p \in E$ that significantly deviate from the rest of data with respect to the spatial component, i.e.

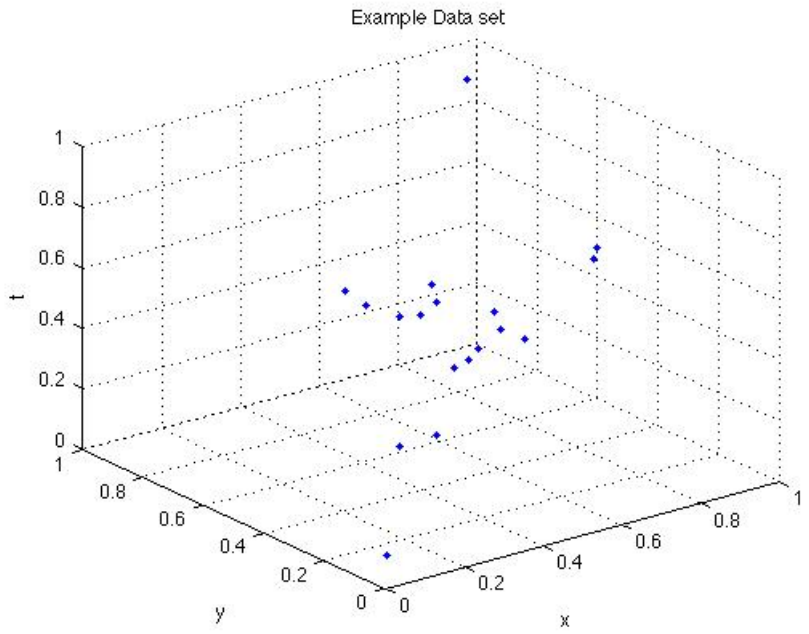


Figure 5.2: Example dataset

Example 1. $O_s = \{(0.95, 0.55, 0.50), (1, 0.60, 0.50),$

$(0.01, 0.01, 0.1), (0.9, 0.9, 0.95)\}$

O_s set is shown in the Figure 5.3. A 4-*Temporal Outlier Set* $O_t \subseteq E$ is the set of objects $p \in E$ that significantly deviate from the rest of data with respect to the temporal component, i.e.

Example 2. $O_t = \{(0.01, 0.01, 0.1), (0.20, 0.21, 0.3),$

$(0.30, 0.22, 0.3), (0.9, 0.9, 0.95)\}$

O_t set is shown in the Figure 5.4.

If $n = 2$, a 2-*Spatio-Temporal Outlier Set* $O_{s,t} \subseteq E$ is the set of objects $p \in E$ that significantly deviate from the rest of data with respect to the spatial and the temporal component, i.e.

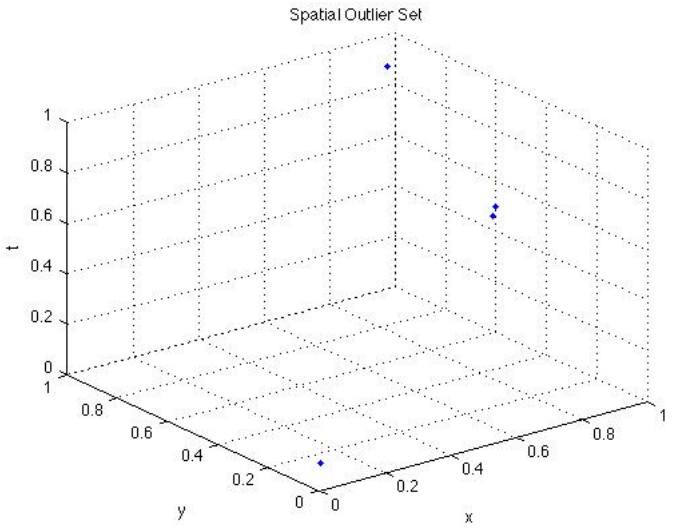


Figure 5.3: Example dataset: 4-Spatial Outlier Set

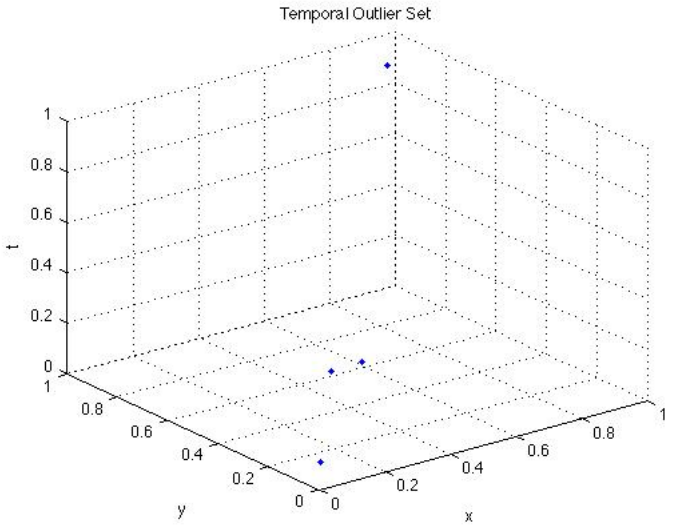


Figure 5.4: Example dataset: 4-Temporal Outlier Set

Example 3. $O_{s,t} = \{(0.01, 0.01, 0.1), (0.9, 0.9, 0.95)\}$

$O_{s,t}$ set is shown in the Figure 5.5.

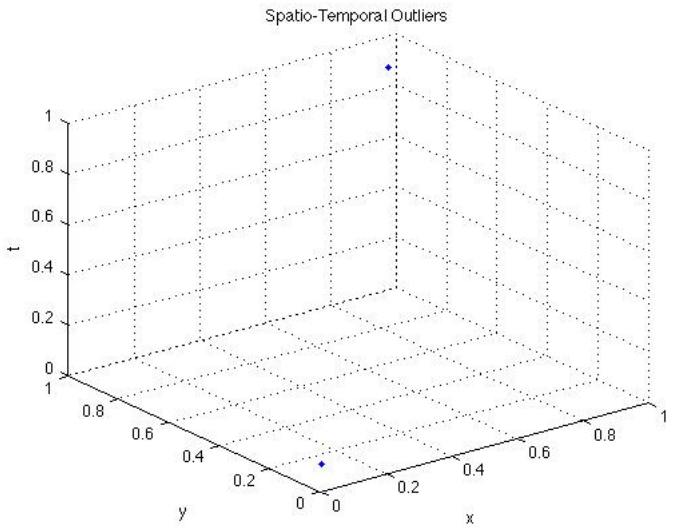


Figure 5.5: Example dataset: 2-Spatio-Temporal Outlier Set

5.3.2 Kernel Set

Let us now define *Kernel Set* $K \subseteq U$ as a selected subset of the universe U, containing the Outlier Set, that characterizes the overall dataset. Intuitively, this set is a subset of objects of U that maintains the general structure of the universe U. The Kernel Set is built by construction, in an iterative way, adding each object having specific properties.

Definition 15. *Given U and two integers $n > 0$, $k > 0$ (number of nearest neighbors), $d(U)$ a measure defined on U, the Kernel Set K is built by adding each object $p \in U$ such that one of the following properties holds:*

1. $d_p(U) \geq \tau$

2. *if* $d_p(U) < \tau$ *then* $\exists q \in U$ *such that* $p \in NN^k(q)$ *and* $d_q(U) < \tau$ *and* $d_q(K - \{p\}) \geq \tau$

where $NN^k(q)$ *is the set of* k*-nearest neighbors of* q *and* $d(K)$ *is the restriction of* $d(U)$ *on* $K \subseteq U$*.*

The definition 15 states that the objects that belong to the **Kernel Set** are:

1. object p for which $d_p(U) \geq \tau$ and hence belongs to *n-Outlier Set*.
2. object p that, even if $d_p(U) < \tau$, is one of the nearest neighbors of an object q for which $d_q(U) < \tau$ and $d_q(K - \{p\}) \geq \tau$.

The second property states that, once these objects p have been added to K, the measure of the object q became smaller than τ both in U and in K. Otherwise, the global structure of the dataset should be altered.
The measure of a given object computed in K is an upper bound of the measure computed in U because some objects belonging to U and not to K could be close to the given object. Also the *kernel Set* is built for the Example dataset in the Example 4:

Example 4. $K = \{(0.01, 0.01, 0.1), (0.9, 0.9, 0.95),$

$(0.95, 0.55, 0.5), (1.0, 0.6, 0.5), (0.2, 0.21, 0.3), (0.3, 0.22, 0.3),$

$(0.3, 0.16, 0.55), (0.35, 0.15, 0.6), (0.15, 0.26, 0.76), (0.16, 0.34, 0.77)\}$

This set is also reported in Figure 5.6. As shown, the Kernel Set contains all the elements of the *Outlier Set*.

The following proposition holds:

Proposition 3. *The measure computed in* K *is an upper bound of the measure computed in* U *such that:*

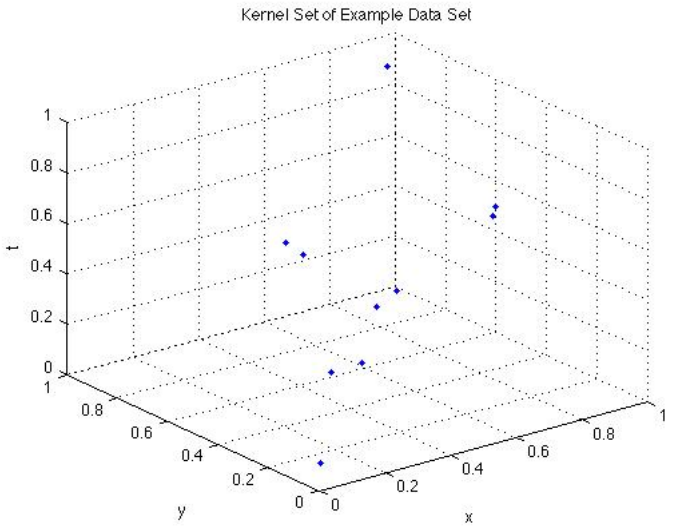


Figure 5.6: Example dataset: Kernel Set

$$d_p(U) \leq d_p(K), \quad \forall p \in U$$

where $d_p(U) = \sum_{j=1}^{k} d(p, N(p, p_j))$ *and* $N(p, p_j)$ *is the j-nearest neighbor of p.*

Proof.

Let be $N(p, p_j)$, *the j-nearest neighbor of p in U.*

Two cases should be highlighted:

Case 1:

If $N(p, p_j) \in U$ *and* $N(p, p_j) \in K$, $\forall j = 1, ..., k$ *then* $d_p(U) = d_p(K)$

Case 2:

It exists an index i such that $N(p, p_i) \in U$ *and* $N(p, p_i) \notin K$

In this case

$d_p(U) = \sum_{j=1,j\neq i}^{k} d(p, N(p, p_j)) + d(p, N(p, p_i))$

$d_p(K) = \sum_{j=1,j\neq i}^{k} d(p, N(p, p_j)) + d(p, N(p, \overline{p}))$

where $\overline{p} \in K$ is the *k-th nearest neighbor* of p in K to keep into account since $p_i \notin K$.

As p_i is one of *k-nearest neighbor* of p in U, the following inequality holds:

$d(p, N(p, p_i)) < d(p, N(p, \overline{p})) \;\Rightarrow\; d_p(U) < d_p(K)$ □

The following proposition is valid:

Proposition 4. *A* ***Kernel Set*** *contains the n-Outlier Set:* $K \supseteq O$.

Proof.

$\forall p \in O : d_p(U) > \tau \Rightarrow p \in K$

The proof is simple since it clearly follows from definition of K. □

5.3.3 Our approach ROSE - Rough Outlier Set Extraction

The pursued approach uses a well-known outlier definition based on k-nearest neighbors [70] to detect the *Outlier Set* and exploits rough set theory to define this set such as a *Rough Outlier Set*.

Theory

For simplicity, let us consider that dataset features are only space and time and let U denote our universe (a normalized spatio-temporal dataset), i.e.

$$\mathrm{U} = \{p_i \equiv (z_{i,1}, z_{i,2}, z_{i,3}) \in [0,1]^3,\ \ \mathrm{i}{=}1,\ ...,\ \mathrm{N}\}$$

The $(z_{i,1}, z_{i,2})$ are spatial components of the i-th object, $z_{i,3}$ is its relative time-stamp. In this case, the attribute set is $A = \{x, y, t\}$, i.e. x and y are cartesian coordinates and t is the temporal component.
Now, we want to describe $O \subseteq U$ (*n-Outlier Subset*) such as

$$< \underline{B}(O), \overline{B}(O) > \ (\textit{Rough n-Outlier Set})$$

where $\underline{B}(O)$ is the *B-Lower approximation* and $\overline{B}(O)$ is the *B-Upper approximation* of *n-Outlier Set* with respect to an attribute subset $B \subseteq A$.
Let us I_B consider the *B-indiscernibility relation* on the universe U:

$$I_B = \{(p_i, p_j) \in U \times U : a(p_i) = a(p_j),\ \forall a \in B\}$$

The equivalence classes $[p_j]_B$ or granules G_j of the partition induced by I_B on U are such that:

$$U = \bigcup_{j=1}^{N} G_j.$$

and

$$G_j \cap G_j = \emptyset, \quad i \neq j.$$

Example
As instance, let us consider the labeled Example dataset.
Spatial Outliers
In this case, we can reduce by means of temporal component, i.e. $B = \{t\}$, then we have the following partition of the universe:

$$I_B = I_{\{t\}} = \{\{p_1, p_2\}, \{p_3, p_9\}, \{p_4\}, \{p_5\}, \{p_6\}, \{p_7, p_8\}, \{p_{10}\}, \tag{5.9}$$

$$\{p_{11}\}, \{p_{12}\}, \{p_{13}\}, \{p_{14}\}, \{p_{15}\}, \{p_{16}\}, \{p_{17}\}, \{p_{18}\}\}$$

In the 5.9, the equivalence classes or granules, induced by relation $I_{\{t\}}$, have been reported.
The concept of *Spatial Outlier* can be defined on the basis of knowledge in $B = \{t\}$. Specifically, the *B-lower approximation* of the *Spatial OutlierSet* O_s, is composed by the granules completely included into O_s, i.e.

$$\underline{B}(O_s) = \{\{p_7, p_8\}, \{p_{17}\}, \{p_{18}\}\}$$

and the *B-upper approximation* is composed by the granules that have non trivial intersection with O_s, i.e.

$$\overline{B}(O_s) = \{\{p_7, p_8\}, \{p_{17}\}, \{p_{18}\}\}$$

In this case, we do not have any added information by the upper approximation.

Temporal Outliers

In this case, we can reduce by spatial components, i.e. $B = \{x, y\}$, getting:

$$I_B = I_{\{x,y\}} = \{\{p_1, p_{12}\}, \{p_2, p_{13}\}, \{p_3\}, \{p_4\}, \{p_5\}, \{p_6\}, \{p_7\}, \\ \{p_8\}, \{p_9\}, \{p_{10}\}, \{p_{11}\}, \{p_{14}\}, \{p_{15}\}, \{p_{16}\}, \{p_{17}\}, \{p_{18}\}\} \tag{5.10}$$

In the 5.10, the equivalence classes or granules, induced by relation $I_{\{x,y\}}$, have been reported.

The concept of *Temporal Outlier* can be equivalently gets on the basis of knowledge in $B = \{x, y\}$. The *B-lower approximation* of the *Temporal Outlier Set* O_t, is composed by the granules completely included into O_t, i.e.

$$\underline{B}(O_t) = \{\{p_{17}\}, \{p_{18}\}\}$$

and the *B-upper approximation* is composed by the granules that have non trivial intersection with O_t, i.e.

$$\overline{B}(O_t) = \{\{p_1, p_{12}\}, \{p_2, p_{13}\}, \{p_{17}\}, \{p_{18}\}\}$$

In this case, the notion of rough set arises; indeed we have some added information by the upper approximation.

Now, coming back to theory for an unlabeled spatio temporal dataset, we use the measure defined in 5.6 as a weight $\overline{\omega}_{G_j}(s, t, i)$, to be assigned to each granule G_j, depending on space, indicated by s, and/or on time, indicated by t and on iteration, indicated by i. The assigned weights are refined, step by step, keeping into account an higher number of objects of the universe.

The *B-Lower approximation* $\underline{B}(O)$ is defined as the set of objects that can be certainly classified as members of the set O on the basis of the knowledge in B, while the objects in the *B-Upper approximation* $\overline{B}(O)$ can only be classified as possible members of O on the basis of the knowledge in B.

In our framework, the *B-Lower approximation* can be defined as follows:

Definition 16. *The B-Lower approximation $\underline{B}_i(O)$ of n-Outlier Set O, at iteration i, is:*

$$\underline{B}_i(O) = \{G_j \subseteq U : \overline{\omega}_{G_j} > \tau_i\} \tag{5.11}$$

where τ_i is a threshold, at iteration i, as

$$\tau_i = inf\ \{max_1^i\ (\overline{\omega}_{G_j}, \overline{\omega}_{G_k}), ..., max_n^i\ (\overline{\omega}_{G_j}, \overline{\omega}_{G_k})\},\ \forall\ G_j, G_k \subseteq U \tag{5.12}$$

Similarly, the following definition holds:

Definition 17. *The B-Upper approximation $\overline{B}_i(O)$ of n-Outlier Set O, at iteration i, is:*

$$\overline{B}_i(O) = \{G_j \subseteq U : \overline{\omega}_{G_j} > \overline{\tau}_i\} \tag{5.13}$$

where $\overline{\tau}_i$ is

$$\overline{\tau}_i = \tau_{i-1},\ \forall\ \ i >= 2 \tag{5.14}$$

The threshold τ_1 is computed as the minimum value among the n higher values of weights assigned to the granules at first iteration, then, at second iteration, τ_2 will be the new minimum value among the new n higher values of weights re-assigned to the granules at second iteration and $\overline{\tau}_2 = \tau_1$.
The iterative procedure will stop when the following convergence criterion will be satisfied:

Lemma 1. *The construction of the B-Lower approximation $\underline{B}(O)$ or the B-Upper approximation $\overline{B}(O)$ of an n-Outlier Set O converges if it exists an index k such that the*

threshold does not vary anymore then the lower and upper approximations have been reached, i.e.

$$if \quad \overline{\tau}_k = \tau_k \quad then \quad \underline{B}_k(O) = \overline{B}_k(O) \tag{5.15}$$

Proof.

Since $\underline{B}_k(O) = \{G_j \subseteq U : \overline{\omega}_{G_j} > \tau_i\}$ and $\overline{B}_k(O) = \{G_j \subseteq U : \overline{\omega}_{G_j} > \overline{\tau}_i\}$,

we would to prove that:

$$\underline{B}_k(O) = \overline{B}_k(O) \; iff \; 1) \; \underline{B}_k(O) \subseteq \overline{B}_k(O) \quad and \quad 2) \quad \overline{B}_k(O) \subseteq \underline{B}_k(O) \tag{5.16}$$

By definition

$$\forall G_j \subseteq \underline{B}_k(O): \quad \overline{\omega}_{G_j} > \tau_k$$

and by hypothesis, $\exists \;\; k: \quad \overline{\tau}_k = \tau_k$, then:

$$\forall G_j \subseteq \underline{B}_k(O): \quad \overline{\omega}_{G_j} > \overline{\tau}_k \Longrightarrow G_j \subseteq \overline{B}_k(O)$$

Thus $\underline{B}_k(O) \subseteq \overline{B}_k(O)$.

Similarly, by definition,

$$\forall G_j \subseteq \overline{B}_k(O): \quad \overline{\omega}_{G_j} > \overline{\tau}_k$$

and by hypothesis, $\exists \;\; k: \quad \overline{\tau}_k = \tau_k$, then:

$$\forall G_j \subseteq \overline{B}_k(O): \quad \overline{\omega}_{G_j} > \tau_k \Longrightarrow G_j \subseteq \underline{B}_k(O)$$

and thus $\overline{B}_k(O) \subseteq \underline{B}_k(O)$ as required.

□

Hence, the *Rough n-Outlier Set* is represented by:

$$< \underline{B}_{k-1}(O), \overline{B}_{k-1}(O) > \tag{5.17}$$

In case of $B = A$, (every attribute is considered), the granules are:

$$\forall p_j \in U : \{p_j\} \equiv G_j \qquad \forall j = 1, ..., N \tag{5.18}$$

so both spatial and temporal components are taken into account.

ROSE Algorithm

The *Rough Outlier Set Extraction (ROSE) Algorithm* is designed to receive in input the universe U, the number k of nearest neighbors and the number n of outliers to find. The output of the (iterative) procedure is the *Rough Outlier Set* (Upper, Lower Approximation and Negative Region).

The algorithm selects, at each step, a small subset of objects, called *WorkingSet*, from the overall dataset U. At this aim, *ExtractElements* extracts a number of elements equal to a fixed percentage of the cardinality of U that has to be greater than k.
The following main computations are executed. For all selected objects, the procedure computes the Euclidean distances among the objects in the *WorkingSet* and all the objects of U, considering the spatial components, the temporal components or both of them (general case $B = A$) depending upon the chosen attribute subset B with respect to the *Rough Outlier Set* has been calculating. In the following pseudocode, algorithm ROSE related to the general case has been shown. *UpdateUpperApprox* and *UpdateLowerApprox* at first iteration, create the same set of n top outliers at that step, i.e. the n objects that have an associated measure higher than the others. Then, at next iterations, *UpdateUpperApprox* and *UpdateLowerApprox* compute the Lower and Upper approximation of *Rough Outlier Set*, using the τ (computed by *LowerWeight*) and τ_prev thresholds as respectively defined in 5.12 and 5.14. At each iteration i, the pruning strategy selects objects from U that have their measure under the computed threshold in order to build the Negative Region.

1: *begin* **ROSExtraction (U, n, k)**
2: $UpperOutlierSet = null$
3: $LowerOutlierSet = null$
4: $w_{s,t,k}(q) = 0$
5: $\tau\_prev = 0$
6: $\tau = 0$
7: $WorkingSet = ExtractElements(U)$
8: **while** $(WorkingSet! = null)$ **do**
9: **for** $p \in U$ **do**
10: **for** $q \in WorkingSet$ **do**
11: **if** $(LowerOutlierSet == null\ and\ UpperOutlierSet == null)\ or\ (w_{s,t,k}(q) \geq \tau\_prev))$ **then**
12: $d_s(p,q) = CalculateSpDistance(p,q)$
13: $d_t(p,q) = CalculateTempDistance(p,q)$
14: $BuildTreeKNN(p,q,d_s,d_t,k)$
15: **else**
16: $AddNegativeRegion(p)$
17: **end if**
18: **end for**
19: **end for**
20: **for** $q \in WorkingSet$ **do**
21: $w_{s,t,k}(q) = CalculateWeight(q)$
22: $UpperOutlierSet = UpdateUpperApprox(\tau\_prev, n, w_{s,t,k}(q))$
23: $LowerOutlierSet = UpdateLowerApprox(\tau, n, w_{s,t,k}(q))$
24: **end for**
25: $\tau = LowerWeight(UpperOutlierSet)$
26: **if** $(\tau! = 0)$ **then**
27: $\tau\_prev = \tau$
28: **end if**
29: $U = U - WorkingSet$
30: $WorkingSet = ExtractElements(U)$
31: **end while**
32: **return** *Rough Outlier Set*
33: *end* **ROSExtraction ()**

KSE - kernel Set Extraction

Another procedure allows to build also the *kernel Set*, that is a selected subset of objects belonging to the universe U that preserves the general data structure of the universe. The advantage of working with *kernel Set* instead of original universe U is related to the set cardinality. To this aim, in this section, we propose the comparison between the obtained results, in terms of Rough Outlier Set, once computed from the entire universe U and another time computed from the Kernel Set K.

The *kernel Set* is a significative subset of the universe U with the following properties:

- computational benefits: *Kernel Set* is a subset with lower cardinality than U
- the "same results" in terms of *Rough Outlier Set* can be obtained using *kernel Set* instead of U
- *kernel Set* can be considered as the model learned during a training phase.

Let us start to prove the following Proposition:

Proposition 5. *The Outlier Set O_K, computed starting from kernel Set K is a superset of O computed from U:*

$$O_K \supseteq O$$

Proof.

Let be O the *n-Outlier Set* computed from U:

$$O = \{p_1, ..., p_n, p_{n+1}, ..., p_{\overline{n}} \in U \ / \ d_{p_1}(U) \geq ... \geq d_{p_n(U)} = d_{p_{n+1}}(U) .. = d_{p_{\overline{n}}}(U) >$$

$$d_{p_j}(U) \ \ \forall j = \overline{n}+1, ..., N\}$$

where $d_{p_i}(U) = \sum_{j=1}^{k} d(p_i, N(p_i, p_j)) \ \ \forall i = 1, ..., N$ is defined and computed on U.

We want to prove that:

$$\{p_1, ..., p_n, p_{n+1}, ..., p_{\overline{n}}\} \in O \textit{ implies that } \{p_1, ..., p_n, p_{n+1}, ..., p_{\overline{n}}\} \in O_k \tag{5.19}$$

By Proposition 3, the following inequality holds:

$$d_p(U) \leq d_p(K), \quad \forall p \in U$$

and in particular:

$$d_{p_i}(U) \leq d_{p_i}(K), \quad \forall i = 1, ..., \overline{n}$$

By definition of *n-Outlier Set*

$$d_{p_i}(U) \geq \ \tau, \ \forall i = 1, ..., \overline{n}$$

Thus:

$d_{p_i}(K) \geq \ \tau \ \forall i = 1, ..., \overline{n}$ implies $\{p_1, ..., p_n, p_{n+1}, ..., p_{\overline{n}}\} \in O_k$,

letting the thesis to hold. □

5.3.4 Dependency Rule Generation

In this section, we want to demonstrate how dependency rules can support the user to select a significative subset of attributes that play a significative role describing a concept.
To this aim, we are going to face our unsupervised problem, as follows:

- subdividing our dataset in training and test set;

- running our detection method on training set;
- applying the general methodology for Rough Set Rule Generation to the targeted data;
- using the extracted Rules to classify an unseen object in the test set.

Let us the general problem ST-Outlier Detection and let us consider

$$S =< U, A >$$

our information system, where U is our universe of the discourse and A is our attribute set.
As example, let us consider

$$U = \{p_1, p_2, ..., p_{16}\}$$

a dataset that consists of 16 samples coming from a video tracking with the attribute set

$$A = \{x, y, t, p, a, d\}$$

where x, y are the spatial coordinates of moving centroids (a man walking), t is the temporal component, p and a are the perimeter and the area of minimum bounding box surrounding the tracked object, respectively.
In the figure 5.7, the 3D-projection (x, y, t) of the dataset has been shown.

The ST-Outliers are the red circled positions in the figure.
First of all, we want to identify the d-reducts for our case. As described above, the d-reducts are the minimal set of attributes able to induce the same partition on the domain as done by A, the entire set of attributes.
To this aim, our attribute set $A = \{x, y, t, p, a, d\}$ could be written as $A = C \bigcup \{d\}$ where $C = \{x, y, t, p, a\}$ is the subset of condition attributes and $\{d\}$ is our decision attribute. The value set V_d of $\{d\}$ is the following:

$$V_d = \{d_1, d_2\},$$

i.e. Inlier and Spatio-Temporal Outlier respectively.
U may be consequently partitioned as:

$$U = U_1 \bigcup U_2$$

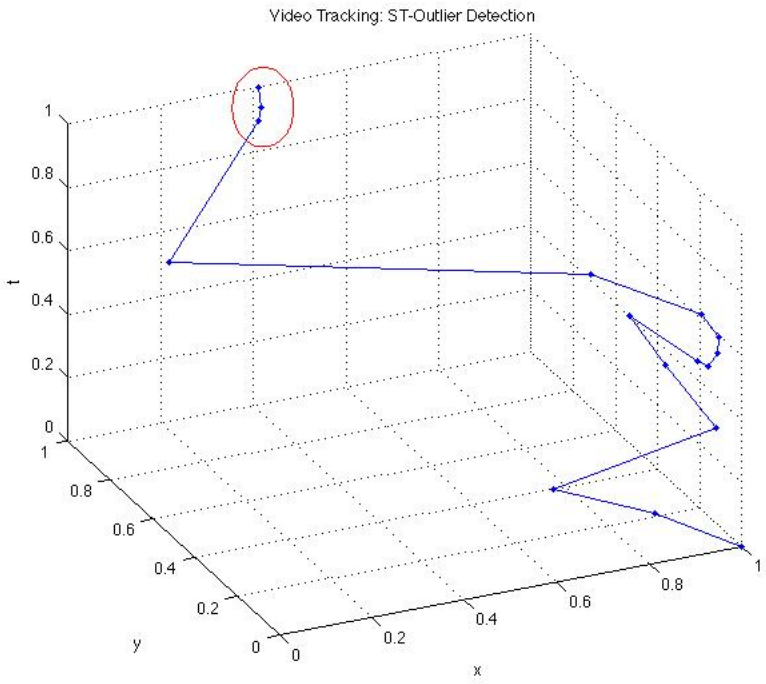


Figure 5.7: dataset Video Tracking

and

$$A = C \bigcup \{d_i\}, \quad i = 1, 2$$

in order to obtain two different decision tables:

$$S_i =< U_i, A_i >, \quad i = 1, 2$$

as depicted in table 5.1.

The discernibility matrix for each decision table can be computed according to (5.2.2). Table 5.2 show the discernibility matrix $M_{ST-Outlier}(C)$ and table 5.3 show the discernibility matrix $M_{Inlier}(C)$, for the condition attribute set C.

The discernibility function for the concept "ST-Outlier" is:

$$f_{ST-Outlier} = (x \vee y \vee t) \wedge (x \vee y \vee t) \wedge (x \vee y \vee t)$$

	x	y	t	p	a	**Decision**
p_1	1.00000000	0.08264463	0.00000000	0.00578035	0.00000000	Inlier
p_2	0.73457944	0.11570248	0.05047319	0.32947977	0.19261948	Inlier
p_3	0.59626168	0.23966942	0.10094637	0.42774566	0.27876588	Inlier
p_4	0.80747664	0.01652893	0.35331230	0.40462428	0.22940109	Inlier
p_5	0.80934579	0.26446281	0.40378549	0.23699422	0.13067151	Inlier
p_6	0.80934579	0.44628099	0.45425868	0.12138728	0.07065941	Inlier
p_7	0.81121495	0.11570248	0.50473186	0.34682081	0.20859044	Inlier
p_8	0.79439252	0.01652893	0.55205047	0.68786127	0.75111918	Inlier
p_9	0.80560748	0.00000000	0.60252366	0.61849711	0.61705989	Inlier
p_{10}	0.80747664	0.00000000	0.65299685	0.41618497	0.23593466	Inlier
p_{11}	0.79065421	0.04132231	0.70347003	0.40462428	0.25045372	Inlier
p_{12}	0.66355140	0.23966942	0.75394322	0.60115607	0.45226860	Inlier
p_{13}	0.00000000	0.50413223	0.85488959	1.00000000	1.00000000	Inlier

	x	y	t	p	a	**Decision**
p_{14}	0.33831776	0.98347107	0.90536278	0.00000000	0.00290381	ST-Outlier
p_{15}	0.34018692	0.97520661	0.95268139	0.00000000	0.00290381	ST-Outlier
p_{16}	0.34392523	1.00000000	1.00000000	0.00000000	0.00290381	ST-Outlier

Table 5.1: Spatio-Temporal Outlier Detection: Decision Tables by Different Decision Attribute Values

	p_{14}	p_{15}	p_{16}
p_{14}		x, y, t	x, y, t
p_{15}			x, y, t
p_{16}			

Table 5.2: Spatio-Temporal Outlier Detection: Discernibility matrix $M_{ST-Outlier}(C)$

The dependency rules extracted by its disjunctive normal form represent the concept of "ST-Outlier".

$$f_{ST-Outlier} \leftarrow x \wedge y$$
$$f_{ST-Outlier} \leftarrow x \wedge t$$
$$f_{ST-Outlier} \leftarrow y \wedge t$$

Similarly, we can find the disjunctive normal form by which the dependency rules are extracted and that represent the concept of "Inlier".

5.4 Experimental Results and Discussion

The testing phase on real and synthetic datasets provided interesting results. In this section, the tests are primarily based on the School Buses dataset [35], described in the Section 4.3.2. Indeed, it is a real dataset that, unlike Tracking and Complex that are small and synthetic datasets, possesses a quite large data amount that allows to appreciate better the usefulness of boundary. However, for completeness' sake, we also report the test results on Complex9_RN8_time Dataset, described in the Section 4.3.3.

5.4.1 School Buses dataset: S-Rough representation of Outlier Set

Let U denote the spatio-temporal normalized School Buses dataset

$$\mathrm{U} = \{p_i \equiv (z_{i1}, z_{i2}, z_{i3}) \in [0,1]^3,\ i = 1, ..., N\}$$

where $(z_{i,1}, z_{i,2})$ are cartesian coordinates of the i–th object, $z_{i,3}$ is the relative time-stamp. Let $< U, A >$ be the information system, with the attribute set $A = \{x, y, t\}$, i.e. x and y are the spatial components and t is the temporal component.
We want to describe $O \subseteq U$ $(Outlier\ \ Subset)$ such as

$$< \underline{B}(O), \overline{B}(O) > \quad (Rough\ Outlier\ Subset)$$

where $B \subseteq A$ is constituted by the spatial attributes, (x, y). Selecting only spatial components, in the following, the results of selected iterations, an *intermediate* step, the *last*-1 and the *last* one have been shown.

S-Rough Outlier Set: Intermediate Iteration

In this section, we show the Lower, Upper Approximation and Boundary at an intermediate step.
In Figure 5.8 the lower approximation has been shown. In Figure 5.9 the upper approximation has been shown, while in Figure 5.10 the lower approximation with boundary in red color have been shown.

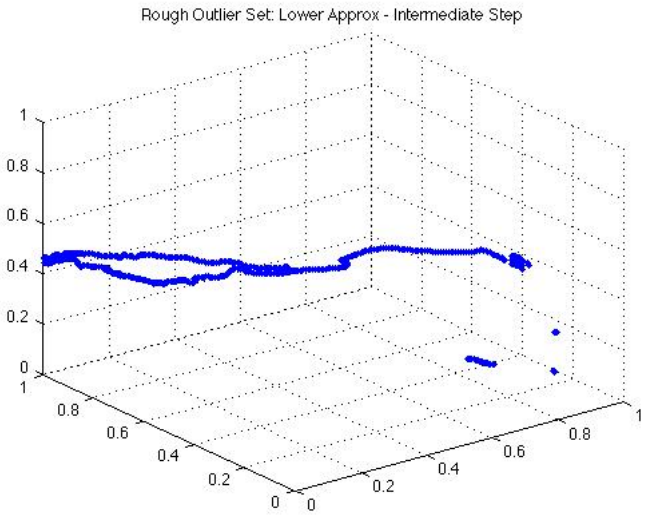


Figure 5.8: Rough Outlier Set: Lower Approximation

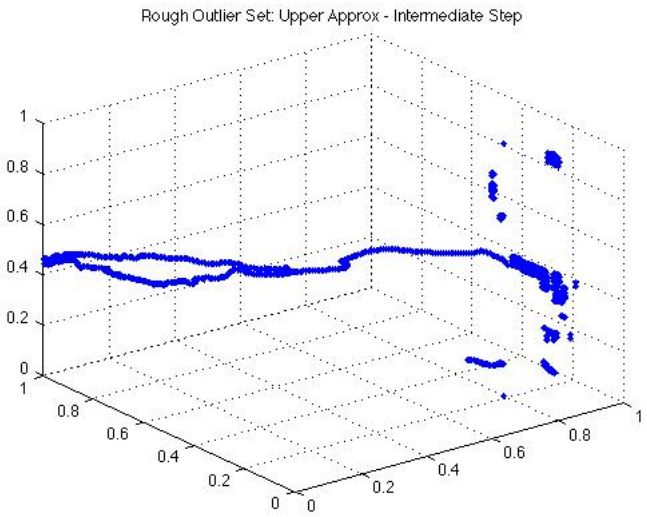


Figure 5.9: Rough Outlier Set: Upper Approximation

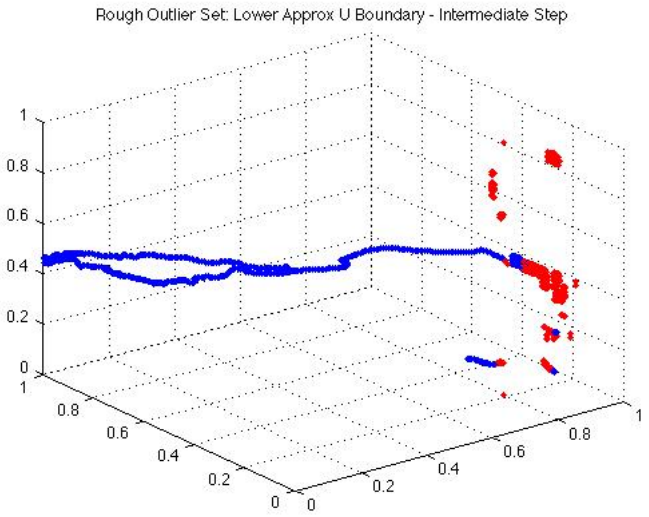


Figure 5.10: Rough Outlier Set: Lower Approximation U Boundary

S-Rough Outlier Set: Last-1 Iteration

In this section, we show the Lower, Upper Approximation and Boundary at last-1 step. In Figure 5.11 the lower approximation has been shown. In Figure 5.12 the upper approximation has been shown and in Figure 5.13 the lower approximation with boundary in red color have been shown.

S-Rough Outlier Set: Last Iteration

In this section, we show the Lower, Upper Approximation and Boundary at last step. In Figure 5.14 the lower approximation has been shown. In Figure 5.15 the upper approximation has been shown and in Figure 5.16 the lower approximation with boundary in red color have been shown. In this last figure, we can see the advantages of keeping into account the boundary. Otherwise, many interesting objects (belonging to the boundary region) should be missed.

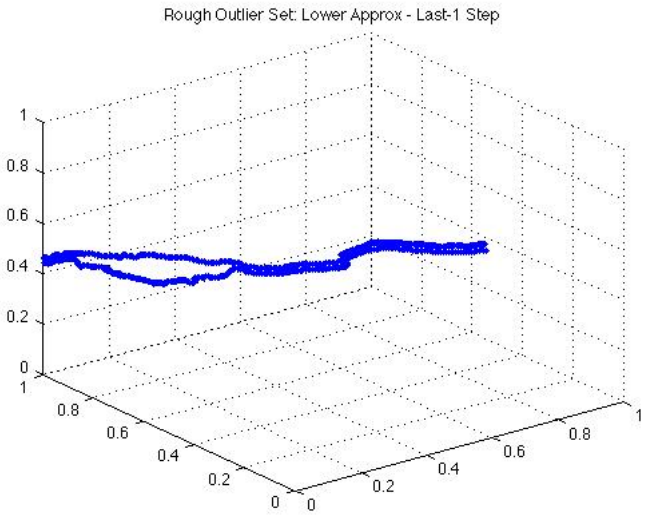


Figure 5.11: Rough Outlier Set: Lower Approximation

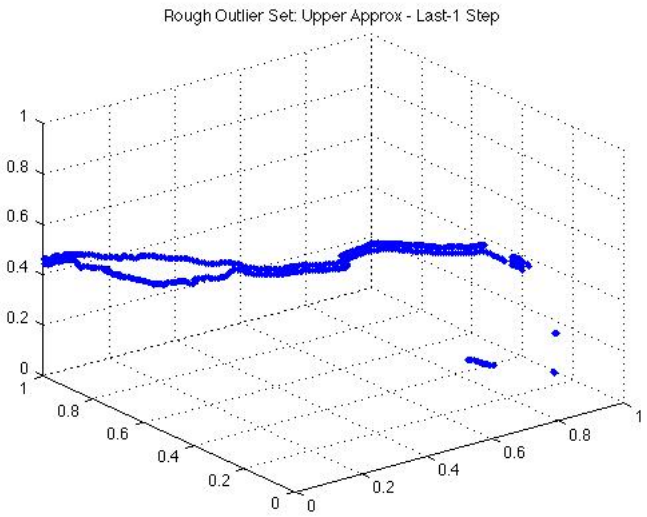


Figure 5.12: Rough Outlier Set: Upper Approximation

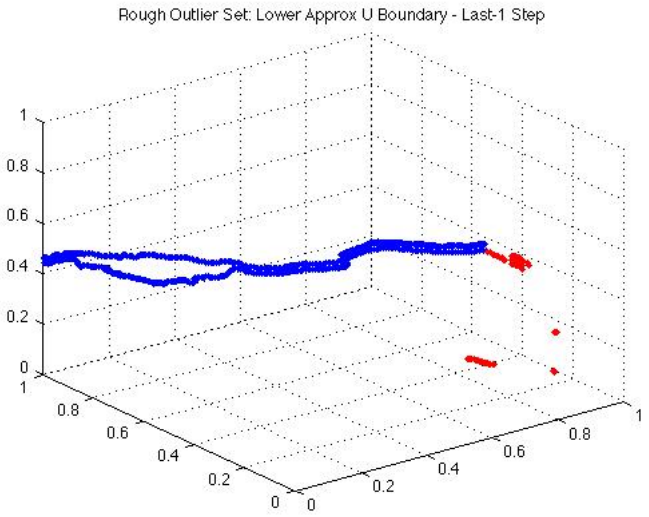


Figure 5.13: Rough Outlier Set: Lower Approximation U Boundary

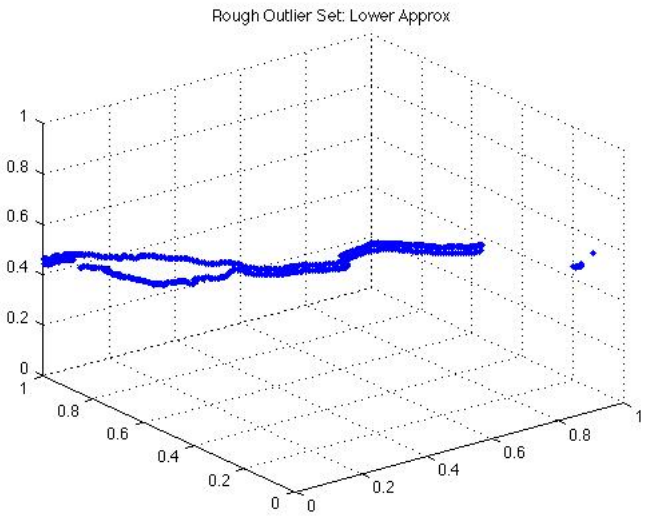


Figure 5.14: Rough Outlier Set: Lower Approximation

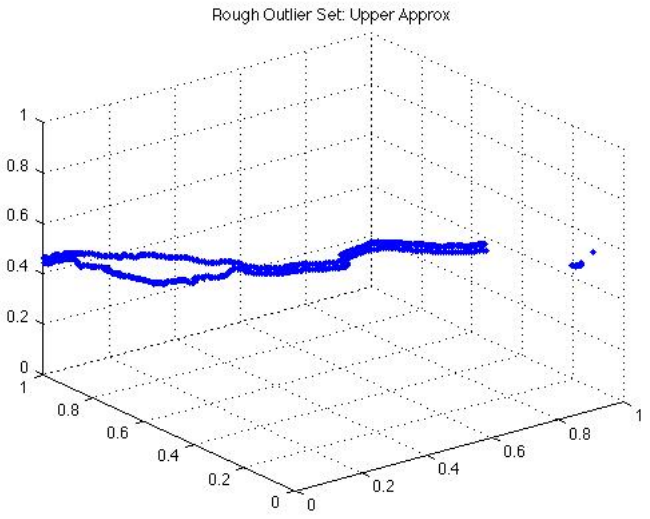


Figure 5.15: Rough Outlier Set: Upper Approximation

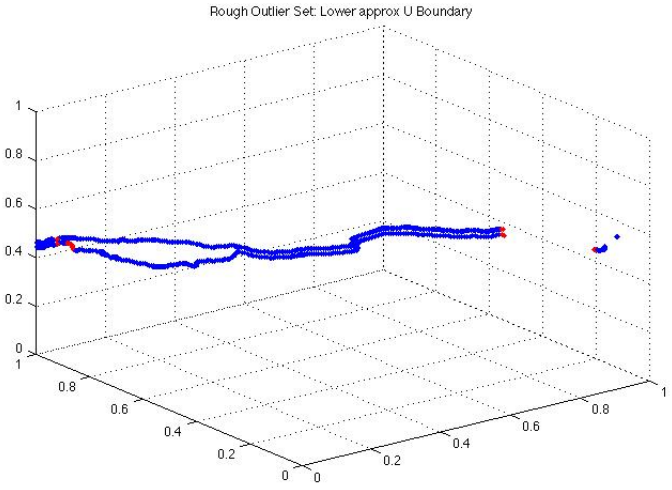


Figure 5.16: Rough Outlier Set: Lower Approximation U Boundary

5.4.2 School Buses Dataset: ST-Rough representation of Outlier Set

Let U denote the spatio-temporal normalized School Buses dataset

$$\mathrm{U} = \{p_i \equiv (z_{i1}, z_{i2}, z_{i3}) \in [0,1]^3,\ i = 1, ..., N\}$$

where $(z_{i,1}, z_{i,2})$ are cartesian coordinates of the i–th object, $z_{i,3}$ is the relative time-stamp. Let $< U, A >$ be the information system, with the attribute set $A = \{x, y, t\}$, i.e. x and y are the spatial components and t is the temporal component.
Now we are considering B=A, so we are looking for *spatio-temporal Rough Outlier Set.* In the following, the results of last iteration have been shown.

ST-Rough Outlier Set: Last Iteration

In this section, we show the Lower, Upper Approximation and Boundary at last step. The spatio-temporal outliers will be more relevant of spatial and temporal outliers (see temporal outliers injected in the Figure 5.17). Hence, the lower approximation includes the most part of spatial and temporal outliers, while the upper approximation includes the remaining part of temporal outliers and some other spatial outliers.

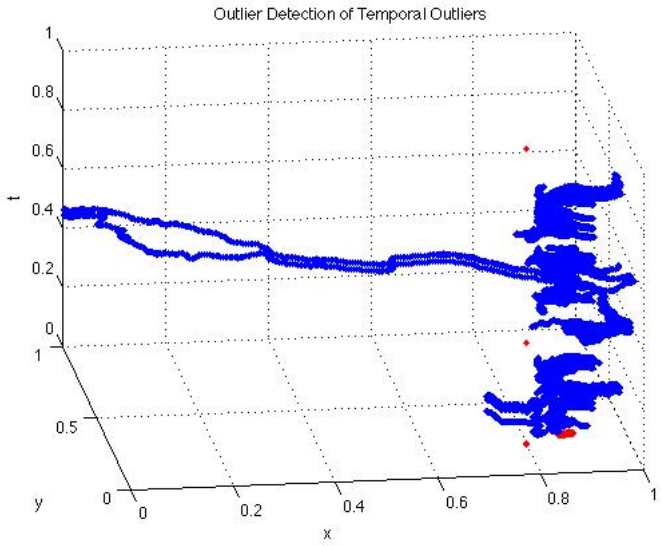


Figure 5.17: Injected Temporal Outliers

In Figure 5.18 the lower approximation has been shown. In Figure 5.19 the upper ap-

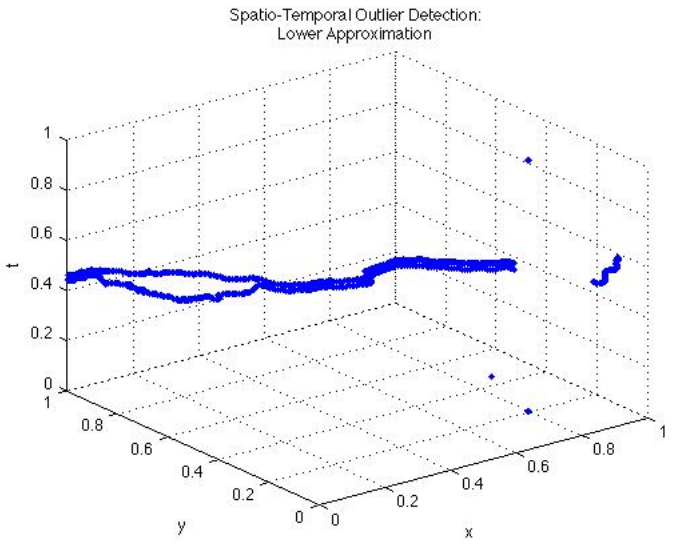


Figure 5.18: ST-Rough Outlier Set: Lower Approximation

proximation has been shown and in Figure 5.20 the lower approximation with boundary in red color have been shown.

5.4.3 School Buses Dataset: Representation of Kernel Set

In this section, we report the tests aimed at demonstrate the use of the Kernel Set. This set is a selected subset, able to describe the original dataset both in terms of data structure and in terms of obtained results. In particular, in the following sections, we want to show the advantages of using this set and the benefits of considering it. At this aim, we show the *Rough Outlier Set* extracted by the universe U and the *Rough Outlier Set* extracted by the Kernel Set. The results will show the advantages of considering this set. Figure 5.21 shows the kernel set of School Buses dataset.

Let be $B \subseteq A$ constituted by the spatial attributes, i.e. (x, y). Selecting only spatial components, in the next subsection, the results of last iteration of the test of Rough Outlier Set Extraction from the Kernel Set have been reported. Thus, we compare these

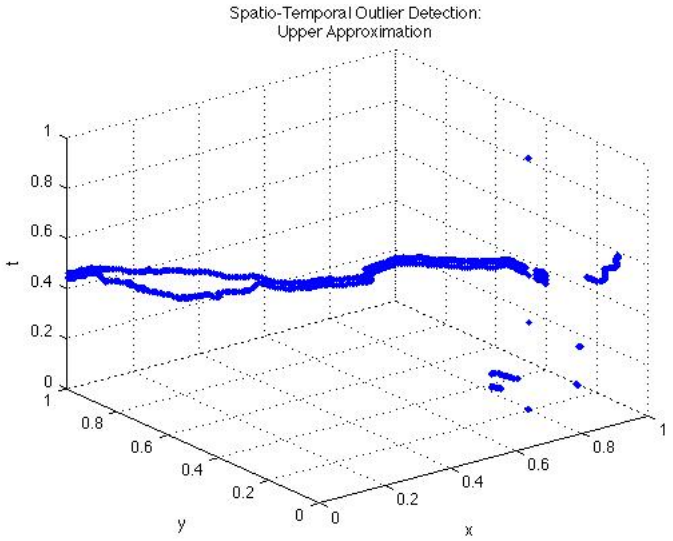


Figure 5.19: ST-Rough Outlier Set: Upper Approximation

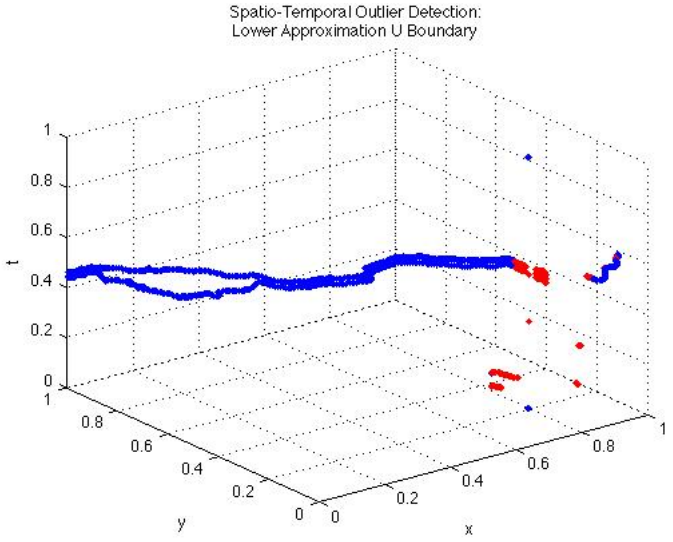


Figure 5.20: ST-Rough Outlier Set: Lower Approximation U Boundary

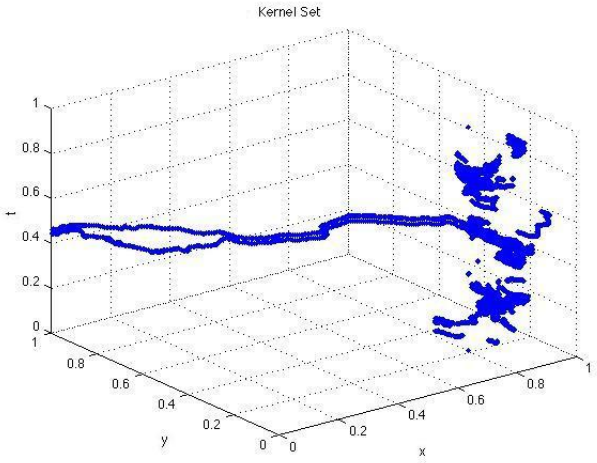


Figure 5.21: School Buses dataset: the Kernel Set

results with the last test of Rough Outlier Set Extraction from the entire universe U, shown in Figure 5.16.

Kernel based - ROSE: Rough Outlier Set from the Kernel Set

Starting from the Kernel S et, the Rough Outlier Set is built by our approach ROSE. Figure 5.22 shows the lower approximation at last iteration, while Figure 5.23 shows the upper approximation, and Figure 5.24 shows the lower approximation with boundary in red color. If we compare Figure 5.16 and Figure 5.24, we can appreciate that the results seem to be quite similar with an interesting computational benefit coming from considering the Kernel set instead of the entire universe U.

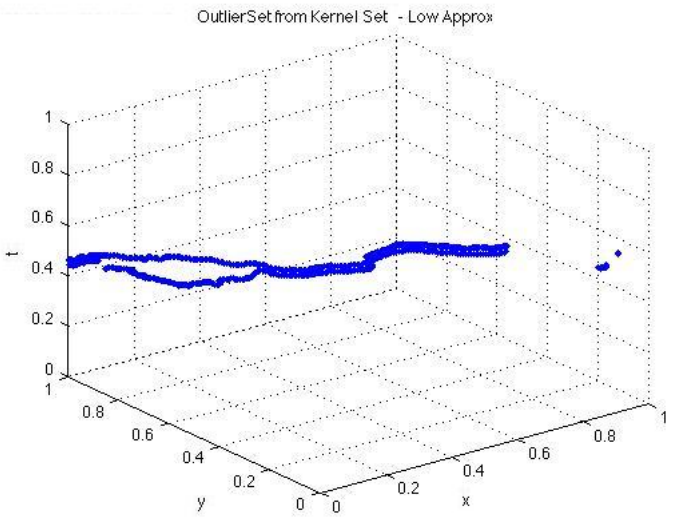


Figure 5.22: Rough Outlier Set: Lower Approximation

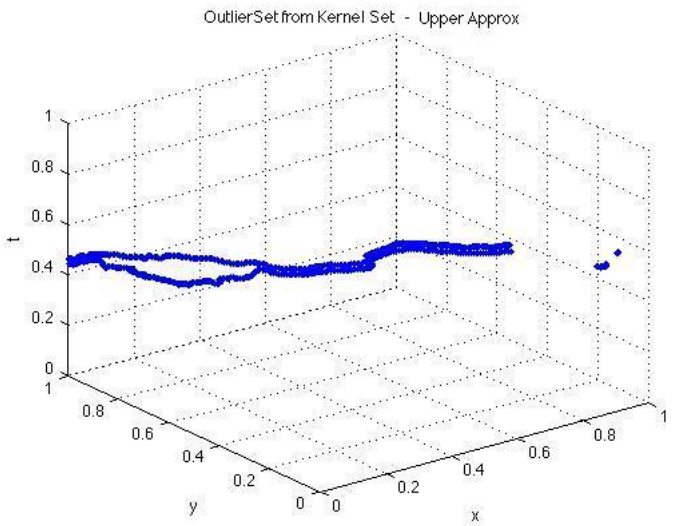


Figure 5.23: Rough Outlier Set: Upper Approximation

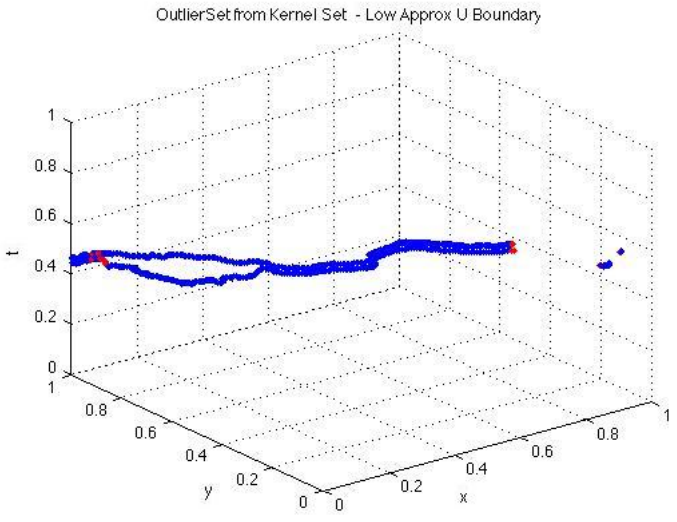


Figure 5.24: Rough Outlier Set: Lower Approximation U Boundary

5.4.4 Complex9_RN8_time Dataset: S-Rough representation of Outlier Set

Let U denote the spatio-temporal normalized Complex9_RN8_time dataset

$$\mathrm{U} = \{p_i \equiv (z_{i1}, z_{i2}, z_{i3}) \in [0,1]^3,\ i = 1, ..., N\}$$

where $(z_{i,1}, z_{i,2})$ are cartesian coordinates of the i–th object, $z_{i,3}$ is the relative time-stamp. Let $< U, A >$ be the information system, with the attribute set $A = \{x, y, t\}$, i.e. x and y are the spatial components and t is the temporal component.
Also, in this case, we want to describe $O \subseteq U$ $(Outlier\ \ Subset)$ such as

$$< \underline{B}(O), \overline{B}(O) > \quad (Rough\ Outlier\ Set)$$

where $B \subseteq A$ is constituted by the spatial attributes, i.e. (x, y). Selecting only spatial components, in the following, the results of *last* iteration have been shown.

S-Rough Outlier Set: Last Iteration

Specifically, the Lower, Upper Approximation and Boundary at last step of *Spatial Rough Outlier Set* are represented and shown in Figure 5.25 where the dataset is shown in light

green color, lower approximation is in blue color and boundaries are shown in red color. As shown, the objects belonging to the boundary (in red color) represent an uncertainty region.

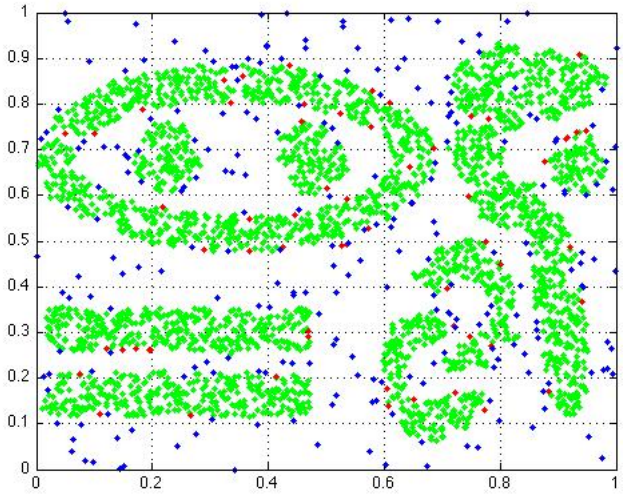


Figure 5.25: Complex9_RN8_time dataset: Last Step - Lower Approximation in blue color and Boundary in red color

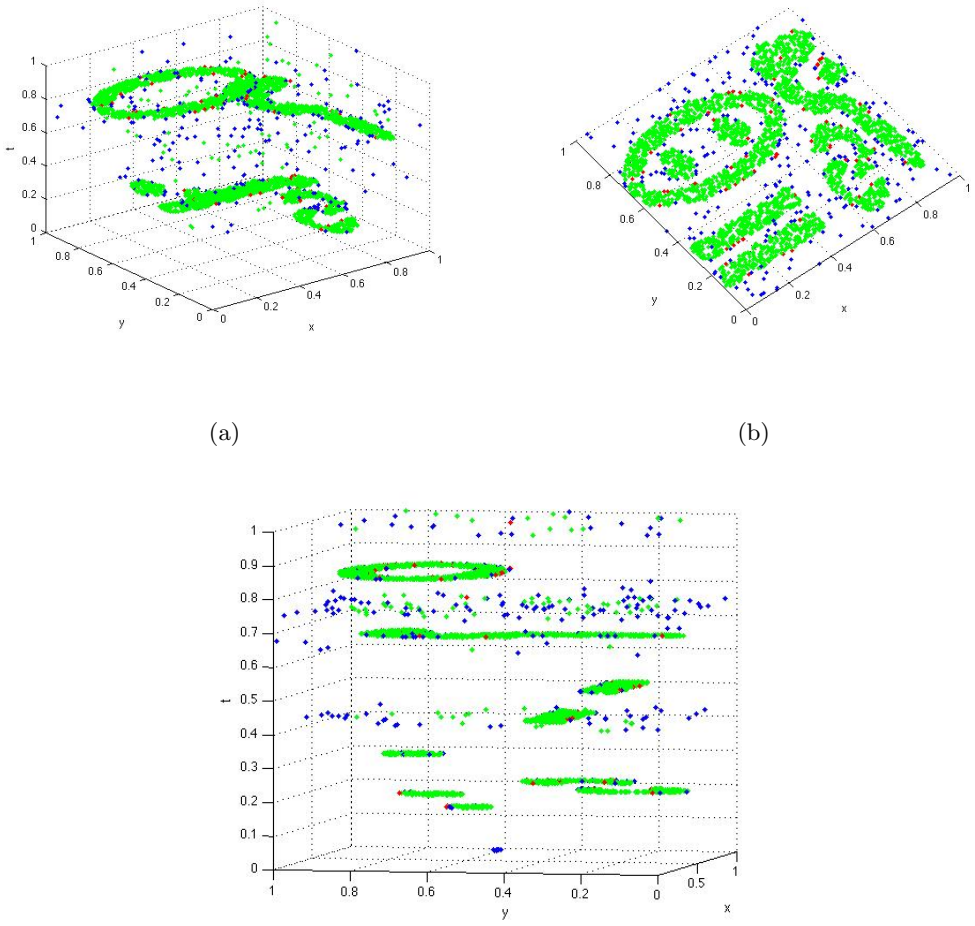
5.4.5 Complex9_RN8_time dataset: ST-Rough representation of Outlier Set

Let $< U, A >$ be the information system, with the attribute set $A = \{x, y, t\}$, i.e. x and y are the spatial components and t is the temporal component. Now we are considering B=A, so we are looking for *spatio–temporal Rough Outlier Set.*
In the following section, the results of *last* iteration have been shown.

ST-Rough Outlier Set: Last Iteration

The spatio-temporal outliers will be more relevant of spatial and temporal outliers. In this section, we show the Lower, Upper Approximation and Boundary of *ST-Rough*

Outlier Set, at last step.



(a) (b)

(c)

Figure 5.26: Complex9_RN8_time dataset: Last Step (a) Lower Approximation in blue color and Boundary in red color (b) A 2D-plotting (c) A different perspective

In Figure 5.26(a) the dataset is shown in light green color, lower approximation is in blue color and boundaries are shown in red color; in Figure 5.26(b) a 2D plotting is reported to verify the spatial detection and in Figure 5.26(c) the figure has been rotated to verify the behaviour with respect to t-component.

5.5 Quantitative Measures and Indices

In this section, we use the performance indeces as introduced by Maji and Pal in [64] such as α index, ρ index and γ index, as well as some measures like the Davies–Bouldin(DB) [19] in order to evaluate the performance of our algorithm compared with some other *rough–fuzzy* clustering algorithms, incorporating the concepts of rough sets. The algorithms [64] are as follows:

- RFCM - Rough Fuzzy C-Means
- RPCM - Rough Possibilistic C-Means
- RFPCM = Rough Fuzzy Possibilistic C-Means.

Let us quickly remind the indices indicated above.

$\alpha-$Index

$$\alpha = \frac{1}{c}\sum_{i=1}^{c}\frac{\omega A_i}{\omega A_i + \tilde{\omega} B_i} \tag{5.20}$$

where A_i is the cardinality of *i-th* cluster lower approximation and B_i is the cardinality of *i-th* cluster boundary, c is the number of clusters, while parameters ω and $\tilde{\omega}$ correspond to the relative importance of lower and boundary regions. The α index represents the average accuracy of c clusters. It represents the average of the ratio of the number of objects in lower approximation to that in upper approximation of each cluster. Indeed, it captures the average degree of completeness of knowledge about all clusters. A good clustering procedure should make all objects as similar to their centroids as possible. The α index increases with an increase in similarity within a cluster, i.e. $0 \leq \alpha \leq 1$.

$\rho-$Index

The ρ index represents the average roughness of c clusters and is defined as follows:

$$\rho = 1 - \alpha = 1 - \frac{1}{c}\sum_{i=1}^{c}\frac{\omega A_i}{\omega A_i + \tilde{\omega} B_i} \tag{5.21}$$

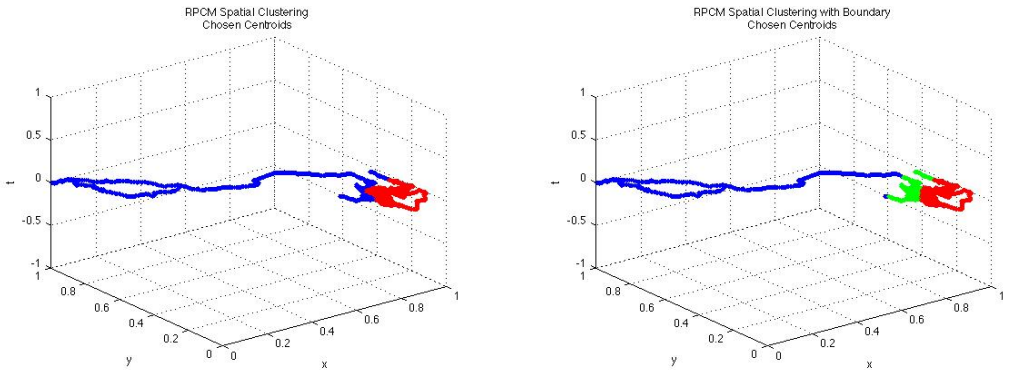
We should remark that the lower is the value of ρ, the better is the over all cluster approximations. As α, $0 \leq \rho \leq 1$. Basically, ρ index represents the average degree of incompleteness of knowledge about all clusters.

$\gamma-$Index

The $\gamma-$Index is the ratio of the total number of objects in the lower approximations of all clusters to the cardinality of the universe of discourse U and is given by:

$$\gamma = \frac{R}{|U|} = \frac{\sum_{i=1}^{c} A_i}{|U|} \tag{5.22}$$

where A_i is the cardinality of i-th cluster lower approximation and U is the universe of our discourse. The γ index basically represents the quality of approximation of a clustering algorithm. To analyze the performance of our proposed algorithm, tests have been done on the School Buses dataset. Figures 5.27 - 5.29 show the results of each algorithm in spatial outlier detection. In the figures (a), the two clusters are drawn in red and blue color after the assignment of the boundary to clusters; while, in the figures (b) the boundary (before the assignment) is drawn in green.

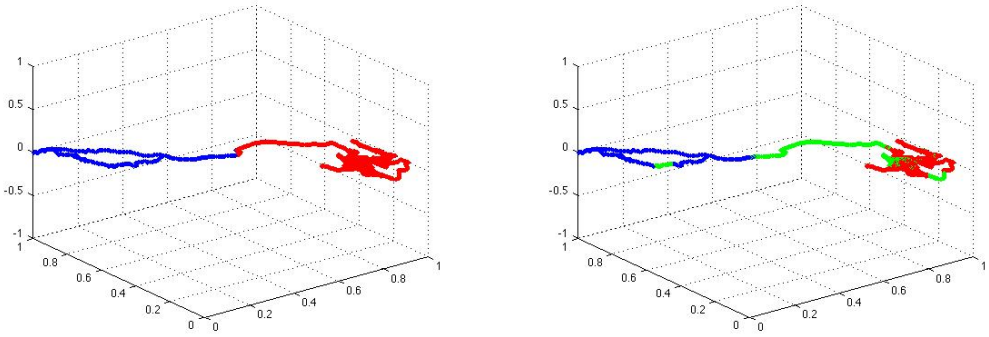


(a) (b)

Figure 5.27: Buses dataset: (a) RPCM Cluster Result (b) RPCM Cluster Result with Boundary

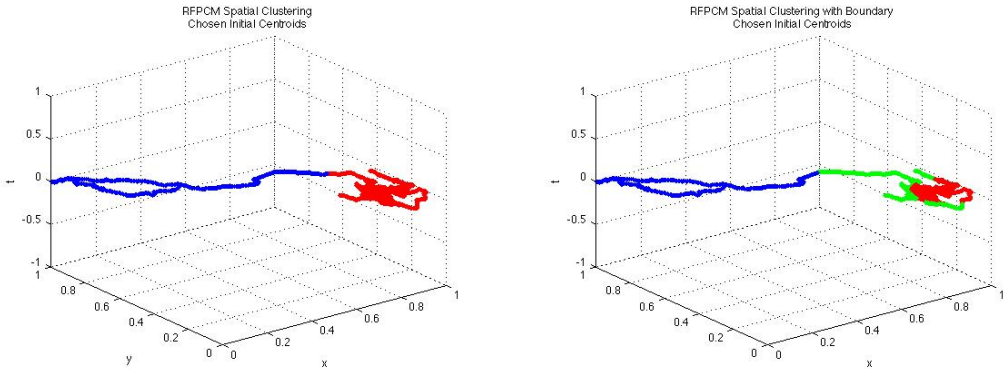
Figures 5.30 - 5.32 show the results of each algorithm in spatio-temporal outlier detection.

The parameters have been set as follows: $c = 2$ (Inlier Cluster and Outlier Cluster), the parameters ω and $\tilde{\omega}$ are equal to 0.5 in order to give the same importance to the lower approximation and to the boundary. Several runs have been made with different initializations and different parameter choices, related to initial centroid choice.These parameters are held constant across all runs. The test shows that the best results are



(a) (b)

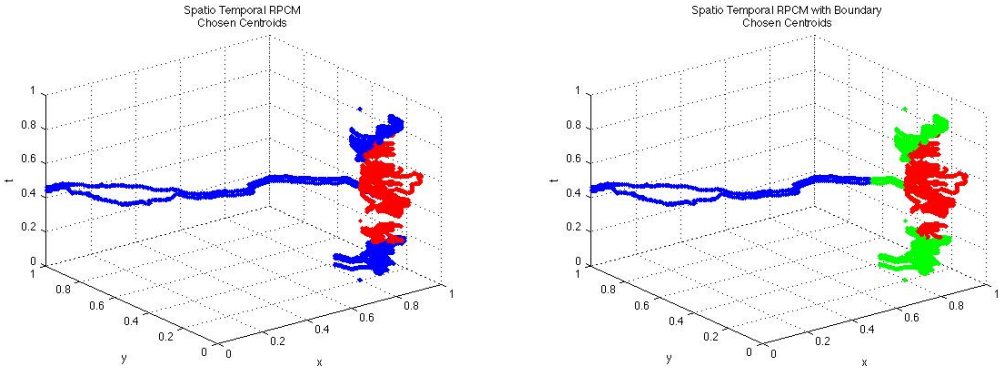
Figure 5.28: Buses dataset: (a) RFCM Cluster Result (b) RFCM Cluster Result with Boundary



(a) (b)

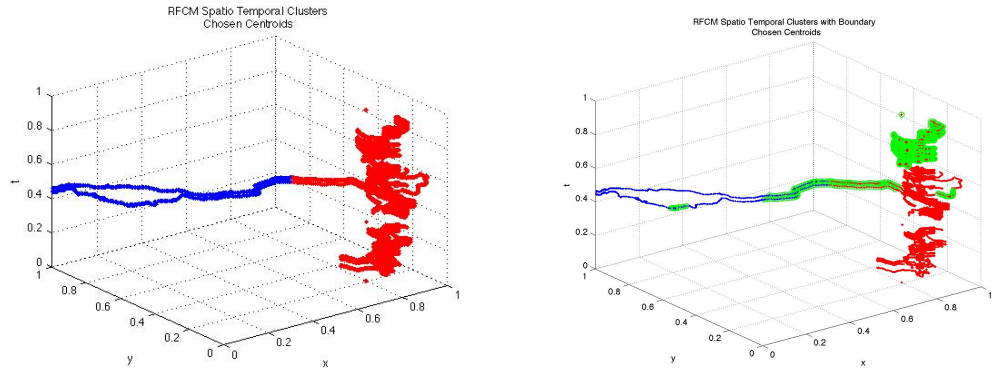
Figure 5.29: Buses dataset: (a) RFPCM Cluster Result (b) RFPCM Cluster Result with Boundary

obtained for particular choices of initial centroids rather than for random choices of initial centroids. So, we report only the final prototypes of the best solution. Table 5.4 and Table 5.5 report the best results obtained using different algorithms for $c = 2$ in case of the same choice of initial centroids for RFCM, RPCM and RFPCM.



(a) (b)

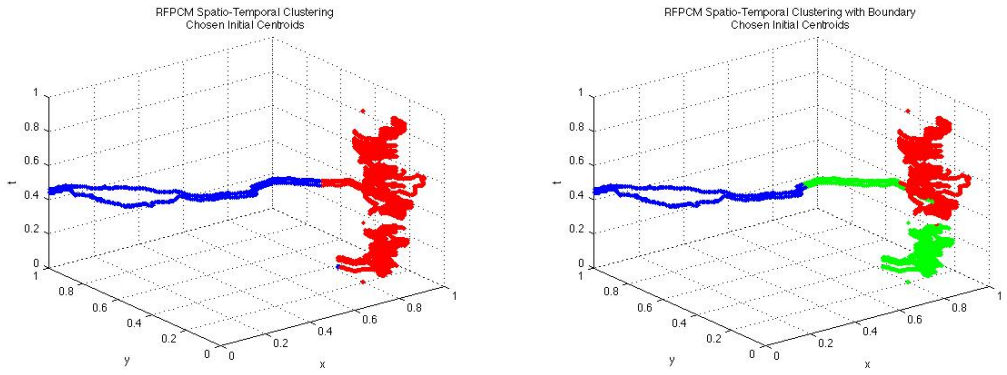
Figure 5.30: Buses dataset: (a) RPCM Cluster Result (b) RPCM Cluster Result with Boundary



(a) (b)

Figure 5.31: Buses dataset: (a) RFCM Cluster Result (b) RFCM Cluster Result with Boundary

Table 5.4 and Table 5.5 compare the performance of these different hybridization rough–fuzzy clustering algorithms with respect to α, ρ, γ and *DB index* in Spatial and Spatio-Temporal Outlier Detection respectively. The results reported in Table 5.4 and Table 5.5 establish the fact that, although these hybridization versions of $c-$means al-



(a) (b)

Figure 5.32: Buses dataset: (a) RFPCM Cluster Result (b) RFPCM Cluster Result with Boundary

gorithm were not designed as outlier detectors, generate good prototypes for $c = 2$. In Spatial Outlier Detection, the RFPCM provides the best result as shown in Figure 5.29; the results of other two versions of rough clustering are quite similar to that of the RFPCM, while in Spatio-Temporal Outlier Detection, the RPCM outperform them as shown in Figure 5.30.
The proposed ROSE algorithm performs better than RFCM, RPCM and RFPCM algorithms, both in terms of qualitative measure and in terms of outliers detected, as shown in figures 5.18 and 5.20.

5.6 Summary

In this chapter, a definition of Outlier Set as Rough Set and a definition of a new set, called Kernel Set, have been provided.
On these two definitions, a Rough Set Based Outlier Detection Method has been developed in order to:

- compute, at each iteration, the lower approximation, upper approximation (and a relative boundary) of the n top outlier set we are looking for;

- exploit the Kernel Set to generate the "same" output results in terms of Rough Outlier Set with computational benefits.

The proposed method has also been compared with some other $rough\text{-}fuzzy$ clustering algorithms, incorporating the concepts of rough sets.

	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
p_1	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a
p_2		$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
		p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a
p_3			$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, t,$	$x, y, t,$
			p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a
p_4				$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, t,$	$x, y, t,$	$y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
				p, a	p, a	p, a	p, a	p, a	p, a	a	p, a	p, a
p_5					$y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
					p, a	p, a	p, a	p, a	p, a	p, a	p, a	p, a
p_6						$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
						p, a	p, a	p, a	p, a	p, a	p, a	p, a
p_7							$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
							p, a	p, a	p, a	p, a	p, a	p, a
p_8								$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
								p, a	p, a	p, a	p, a	p, a
p_9									$x, t,$	$x, y, t,$	$x, y, t,$	$x, y, t,$
									p, a	p, a	p, a	p, a
p_{10}										$x, y, t,$	$x, y, t,$	$x, y, t,$
										p, a	p, a	p, a
p_{11}											x, y, t	x, y, t
											p, a	p, a
p_{12}												x, y, t
												p, a
p_{13}												

Table 5.3: Spatio-Temporal Outlier Detection: Discernibility matrix $M_{Inlier}(C)$

Methods	α **Index**	ρ **Index**	γ **Index**	DB **Index**
ROSE	0.9836	0.0164	0.9987	N.A.
RFCM	0.5448	0.4551	0.9250	0.0736
RPCM	0.4725	0.5274	0.7919	1.1077
RFPCM	0.5645	0.4354	0.9007	0.8983

Legenda:

ROSE = Rough Outlier Set Extraction;

RFCM = Rough Fuzzy C-Means;

RPCM = Rough Possibilistic C-Means;

RFPCM = Rough Fuzzy Possibilistic C-Means.

Table 5.4: Spatial Outlier Detection - Quantitative Evaluation of Algorithms - Chosen Initial Centroids

Methods	α **Index**	ρ **Index**	γ **Index**	DB **Index**
ROSE	0.8941	0.1059	0.9514	N.A.
RFCM	0.3549	0.6450	0.6444	1.8066
RPCM	0.3283	0.6716	0.5914	1.1077
RFPCM	0.3651	0.6348	0.6618	1.3299

Table 5.5: Spatio-Temporal Outlier Detection - Quantitative Evaluation of Algorithms - Chosen Initial Centroids

6 Conclusion, Ongoing and Future Works

6.1 Conclusion

In this thesis, the outlier detection problem in an unlabeled spatio-temporal dataset, i.e. an unsupervised classification problem that do not require targeted data has been faced. In this last chapter, we list the main contributions of this thesis. In this thesis we have tackled this kind of problem from a two different perspectives: the first is a perspective that does not make any assumption on the statistical properties of the data but identify the outliers on the basis of the computation of fully dimensional distances, a data driven approach; the second perspective focuses on approximate representation of knowledge derivable from data by making use of Rough Set Theory, the framework for the construction of approximations of concepts when only incomplete information is available.

The first contribution is a novel *Non Parametric Approach*, called *ST-OutlierDetector*, to face the outlier detection problem in an unlabeled spatio-temporal dataset has been presented. It combines spatial and temporal attributes in order to find out the top outliers. The method has been proved on synthetic and real dataset to be efficient in space and time to detect the spatio-temporal outliers. The strength of this approach is to combine two different features space and time (depending on different aspects) without fixing any spatial and/or temporal criterion. As shown above, the choice of input parameter α is a tricky problem in the general case and the number of outliers required. *ST-OutlierDetector* reports a better classification accuracy respect to the chosen comparing methods.

The second contribution is a *Rough Set Approach* to Outlier Detection, called *ROSE (Rough Outlier Set Extraction)*. This approach is aimed at the representation of the output, the Outlier Set as a Rough Set, i.e. as an imprecise representation of a crisp

set in terms of two subsets, a lower approximation and upper approximation in order to keep into account a level of vagueness that is typical of this kind of problem. The results show that the proposed *ROSE* algorithm performs better than hybridization rough–fuzzy clustering algorithms: RFCM, RPCM and RFPCM, both in terms of qualitative measure and in terms of outliers detected.

Summarizing, the following contributions have been illustrated:

- a Non Parametric Method (also called Combined Approach) "STOutlierDetector"
  - the outlierness degrees highlight spatial and/or temporal aspects depending on the problem to be faced
  - the method is parametric in α letting to give more importance to space or time
- a Rough Set Approach to Outlier Detection "ROSE"
  - the method provides a Rough representation of the Outlier Set
  - the method defines a new set, Kernel Set
    - computational benefits of using Kernel Set
    - the "same results" in terms of *Rough Outlier Set* can be obtained using Kernel Set instead of U
    - Kernel Set can be used as the learned model in a training phase
  - a Kernel based - ROSE Approach.

6.2 Ongoing and Future Works

Future works will explore:

- the applicability of learning strategies for setting input parameter α of Non Parametric Method STOutlierDetector.
- the possibility of extend our approach to the outlier prediction problem
- the applicability of soft computing techniques (such as fuzzy logic) in the spatio-temporal context, in order to define new fuzzy weights as degree of outlierness

- the possibility of define Kernel set as a Rough Set
- the results can be extended for other distance measures, different from Euclidean distance.

Appendix

School Buses Dataset information

The temporal range of School Buses Data set is from 17/10/2000 to 29/10/2001 with many missing dates (also entire months such as: May, July and August of 2001). Moreover, a very different number of entries is provided for each date; it could range from same tens to same thousands. For these tests, we selected from this data set, only the entries dated 2000 as year, i.e. spanning from 24/10/2000 to 14/12/2000. The data set cardinality is about 30000 (about an half of the original one). In the tables 6.2 and 6.2, the entry number of each date, approximatively computed, has been reported. The data set has been shown in figure 4.5 in the chapter 4. Some added temporal outliers can be easily found both for number of neighbors and for the particular date (22/10/2000, 12/11/2000, 22/12/2000). Unlike temporal outliers, no spatial outliers will be added because there are already present several enough evident (ones).

Date	Entry Number	Remarks
22/10/2000	4	Added Outliers
24/10/2000	78	
26/10/2000	94	
27/10/2000	500 (approx.)	
30/10/2000	800 (approx.)	
01/11/2000	450 (approx.)	
02/11/2000	800 (approx.)	
03/11/2000	800 (approx.)	
06/11/2000	600 (approx.)	
07/11/2000	700 (approx.)	
08/11/2000	500 (approx.)	
09/11/2000	600 (approx.)	
10/11/2000	1000 (approx.)	
12/11/2000	4	Added Outliers
13/11/2000	600 (approx.)	
14/11/2000	1000 (approx.)	
15/11/2000	900 (approx.)	
16/11/2000	600 (approx.)	
18/11/2000	400 (approx.)	
19/11/2000	500 (approx.)	

Table .1: Data set: Entry Details by date

Date	Entry Number	Remarks
20/11/2000	600 (approx.)	
21/11/2000	245	
22/11/2000	500 (approx.)	
23/11/2000	1000 (approx.)	
24/11/2000	800 (approx.)	
27/11/2000	800 (approx.)	
28/11/2000	320	
29/11/2000	700 (approx.)	
30/11/2000	700 (approx.)	
01/12/2000	300	
04/12/2000	450 (approx.)	
05/12/2000	1000 (approx.)	
06/12/2000	500 (approx.)	
07/12/2000	800 (approx.)	
08/12/2000	1000 (approx.)	
11/12/2000	700 (approx.)	
12/12/2000	264	
13/12/2000	261	
14/12/2000	169	
22/12/2000	4	Added Outliers

Table .2: Data set: Entry Details by date

	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	**Label**
$z_{1,j}$	0.20	0.21	0.3	2
$z_{2,j}$	0.30	0.22	0.3	2
$z_{3,j}$	0.30	0.16	0.55	0
$z_{4,j}$	0.35	0.15	0.60	0
$z_{5,j}$	0.40	0.14	0.65	0
$z_{6,j}$	0.40	0.16	0.70	0
$z_{7,j}$	0.95	0.55	0.50	1
$z_{8,j}$	1	0.60	0.50	1
$z_{9,j}$	0.35	0.18	0.55	0
$z_{10,j}$	0.50	0.19	0.56	0
$z_{11,j}$	0.25	0.21	0.72	0
$z_{12,j}$	0.20	0.21	0.73	0
$z_{13,j}$	0.30	0.22	0.74	0
$z_{14,j}$	0.34	0.29	0.75	0
$z_{15,j}$	0.15	0.26	0.76	0
$z_{16,j}$	0.16	0.34	0.77	0
$z_{17,j}$	0.01	0.01	0.1	3
$z_{18,j}$	0.9	0.9	0.95	3

Legenda:

0= Inlier 1= Spatial Outlier 2= Temporal Outlier 3= Spatio-Temporal Outlier

Table .3: Example Data set

Bibliography

[1] B. Abraham and G. E. P. Box, Bayesian analysis of some outlier problems in time series. 1979. Biometrika 66, 2, 229-236.

[2] B. Abraham and A. Chuang, Outlier detection and time series modeling. 1989. Technometrics 31, 2, 241-248.

[3] Adam, Nabil R. , Janeja, Vandana Pursnan. *Spatio-Temporal Outlier Detection in Large Databases*. Journal of Computing and Information Technology Inc, 2006.

[4] D. Agarwal, An empirical Bayes approach to detect anomalies in dynamic multidimensional arrays. 2005. In Proceedings of the 5th IEEE International Conference on Data Mining. IEEE Computer Society, 26-33.

[5] D. Agarwal, Detecting anomalies in cross–classified streams: A Bayesian approach. 2006. Knowl. Inform. Syst. 11, 1, 29-44.

[6] C. C. Aggarwal and P. S. Yu, "Outlier Detection for High Dimensional Data", Proc. 2001 ACM SIGMOD Int'l Conf. on Management of Data, pp. 37-46, 2001.

[7] C. C. Aggarwal et al. Fast Algorithms for Projected Clustering ACM SIGMOD Conference Proceedings, 1999

[8] C. C. Aggarwal, P Yu Finding Generalized Projected Clusters in High Dimensional Spaces ACM SIGMOD Conference Proceedings, 2000

[9] C. C. Aggarwal and P. S. Yu, Outlier detection with uncertain data. 2008. In Proceedings of the International Conference on Data Mining (SDM). 483-493.

[10] R. Agrawal, J Gehrke, D. Gunopulos, P. Raghavan Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications ACM SIGMOD Conference Proceedings, 1998

[11] Angiulli, F. and C. Pizzuti, 2005. Outlier mining in large high-dimensional data sets. IEEE Trans. Knowl. Data Eng., 17: 203-215.

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points

to Identify the Clustering Structure", Proc. 1999 ACM SIGMOD IntÕl Conf. on Management of Data (SIGMOD 99), ACM Press, Jun. 1999, pp.49-60.

[13] F. J. Anscombe and I. Guttman, Rejection of outliers. 1960. Technometrics 2, 2, 123-147.

[14] D. Barbara, P. Chen, "Using the Fractal Dimension to Cluster Datasets", In Proceedings of 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Page 260-264, Boston, MA. August 2000.

[15] V. Barnett, The ordering of multivariate data (with discussion). J. Royal Statis. Soc. 1976. Series A 139, 318-354.

[16] V. Barnett, T. Lewis. *Outliers in Statistical Data*. New York: John Wiley; 1994.

[17] S.D. Bay and M. Schwabacher, "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule," Proc. Int'l Conf. Knowledge Discovery and Data Mining (KDD '03).

[18] R. J. Beckman and R. D. Cook Outlier...s. 1983. Technometrics 25, 2, 119-149.

[19] J. C. Bezdek and N. R. Pal, "Some new indexes for cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301-315, Jun. 1988.

[20] A. M. Bianco, M. G. Ben, E. J. Martinez and V. J. Yohai, Outlier detection in regression models with arima errors using robust estimates. 2001. J. Forecast. 20, 8, 565-579.

[21] D. Birant, A. Kut, "Spatio-Temporal Outlier Detection in Large Databases", *Journal of Computing and Information Technology*, vol. 14, no. 4, pp. 291-297, 2006.

[22] T. Bittner, Rough Sets in Spatio-temporal Data Mining. TSDM 2000: 89-104.

[23] G. E. P. Box and G. C. Tiao, Bayesian analysis of some outlier problems. 1968. Biometrika 55, 1, 119-129.

[24] Breunig, M. M., Kriegel, H-P., Ng, R. T. and Sander, J.: LOF: identifying density-based local outliers. In: Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data, Dallas (2000) 93-104

[25] S. D. Byers and A. E. Raftery, Nearest neighbor clutter removal for estimating features in spatial point processes. 1998. J. Amer. Statis. Assoc. 93, 577-584.

[26] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, 2009

[27] Tao Cheng and Zhilin Li, "A Hybrid Approach to Detect Spatial-Temporal Out-

liers", *Proceedings of the 12th International Conference on Geoinformatics*, pp. 173-178, 2004.

[28] Yuming Chen, Duoqian Miao, and Ruizhi Wang " Outlier Detection Based on Granular Computing" *Springer-Verlag Berlin Heidelberg* 2008

[29] Y. Chen et al. "Outlier Detection with the Kernelized Spatial Depth Function", IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 288-305, 2009.

[30] E. Eskin, Anomaly detection over noisy data using learned probability distributions. 2000. In Proceedings of the 17th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., 255-262.

[31] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data," Applications of Data Mining in Computer Security, Kluwer, 2002.

[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. 2nd Int'l Conf. on Knowledge Discovery and Data Mining (KDD 96), AAAI Press, Aug. 1996, pp.226-231.

[33] A. J. Fox, Outliers in time series. 1972. J. Royal Statis. Soc. Series B 34, 3, 350-363.

[34] W. Frawley and G. Piatetsky-Shapiro and C. Matheus, Knowledge Discovery in Databases: An Overview. AI Magazine , Fall 1992, pgs 213-228.

[35] E. Frentzos, K. Gratsias, N. Pelekis, Y. Theodoridis. Nearest Neighbor Search on Moving Object Trajectories. Proc. 9th International Symposium on Spatial and Temporal Databases (SSTD'05), Angra dos Reis, Brazil, August 2005

[36] P. Galeano, D. Pea and R. S. Tsay, 2004. Outlier detection in multivariate time series via projection pursuit. Statistics and econometrics working articles ws044211, Departamento de Estadõstica y Econometrica, Universidad Carlos III.

[37] A.K. Ghosh and P. Chaudhuri, "On Maximum Depth Classifiers," Scandinavian J. Statistics, vol. 32, no. 2, pp. 327-350, 2005.

[38] R. D. Gibbons, Statistical Methods for Groundwater Monitoring. John Wiley & Sons, Inc. 1994.

[39] F. Grubbs, Procedures for detecting outlying observations in samples. 1969. Technometrics 11, 1, 1-21

[40] S. Guha, R Rastogi, K. Shim "CURE: An Efficient Clustering Algorithm for Large Databases" ACM SIGMOD Conference Proceedings, 1998

[41] D. Hand, H. Mannila, P. Smyth: Principles of Data Mining. MIT Press, Cambridge, MA, 2001.

[42] Hawkins, "Identification of Outliers", *Chapman and Hall*,1980.

[43] S. Hawkins, H. He, G. J. Williams and R. A. Baxter, Outlier detection using replicator neural networks. 2002. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery. Springer-Verlag, 170-180.

[44] Z. He, X. Xu, S. Deng (2003) Outlier detection over data streams. In: Proceedings of ICYCS

[45] Hodge V, Austin J: A Survey of Outlier Detection Methodologies. Artificial Intelligence Review 2004 , 22(2):85-126. Publisher Full Text

[46] Huiping Cao, Nikos Mamoulis, and David W. Cheung, "Discovery of Periodic Patterns in Spatio-Temporal Sequences", University of Hong Kong, *IEEE Transactions on Knowledge and Data Engineering*, **19**, 4, April 2007.

[47] Anil K Jain & R.C. Dubes. Algorithm for clustering data. Prentice Hall, New Jersey 1988.

[48] April Jensen, Marina Gavrilova. Neighborhood-based detection of anomalies in high dimension spatio-temporal sensor datasets. Proceedings of the 2004 ACM Symposium on Applied Computing. Nicosia, Cyprus.

[49] Feng Jiang, Yuefei Sui and Cunge " Outlier Detection Based on Rough Membership Function" *Springer-Verlag Berlin Heidelberg* 2006

[50] W. Jin, A.K.H. Tung, and J. Han (2001) Mining top-n local outliers in large databases. In: Proceedings of ACM SIGKDD, pp. 293-298

[51] T. Johnson, I. Kwok, and R.T. Ng. (1998) Fast computation of 2-dimensional depthvcontours. Proceedings of KDD, pages 224-228

[52] R. Jornsten, "Clustering and Classification Based on the L1 Data Depth," J. Multivariate Analysis, vol. 90, no. 1, pp. 67-89, 2004.

[53] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[54] E. M. Knorr and R.T. Ng, "A Unified Notion of Outliers: Properties and Com-

putation", *3-rd International Conference on Knowledge Discovery and Data Mining Proceedings*, pp. 219-222, 1997.

[55] E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets," Proc. 24th Int'l Conf. on Very Large Data Bases, pp. 392-403, 1998.

[56] J. Laurikkala, M. Juhola, E. Kentala (2000) Informal identification of outliers in medical data. In: Proceedings of IDAMAP

[57] M. Lauer, A mixture approach to novelty detection using training data with outliers. 2001. In Proceedings of the 12th European Conference on Machine Learning. Springer-Verlag, 300-311.

[58] J. P. Liu and C. S. Weng, Detection of outlying data in bioavailability-bioequivalence studies. Stat. Med. 1991. 10, 9, 1375-89.

[59] J. Ma and S. Perkins, Online novelty detection on temporal sequences. 2003a. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, 613-618.

[60] J. Ma and S. Perkins, Time-series novelty detection using one-class support vec-

tor machines. 2003b. In Proceedings of the International Joint Conference on Neural Networks. Vol. 3. 1741-1745.

[61] Tuan Trung Nguyen "Outlier Detection: An Approximate Reasoning Approach" *Springer-Verlag Berlin Heidelberg* 2007

[62] T. Odin and D. Addison, Novelty detection using neural network technology. 2000. In Proceedings of the COMADEN Conference.

[63] Pal, S.K., P. Mitra, "Case generation using rough sets with fuzzy representation", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, Mar 2004, pp. 293 - 30.

[64] P. Maji and S.K. Pal, "Rough Set Based Generalized Fuzzy C-Means Algorithm and Quantitative Indices", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 37, n. 6, Dec. 2007

[65] Y. Panatier (1996) Variowin. Software for spatial data analysis in 2D. Springer-Verlag, New York

[66] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos, "LOCI: Fast Outlier

Detection Using the Local Correlation Integral", 19th International Conference on Data Engineering (ICDE'03), 2003

[67] Rough Sets, Theoretical Aspects of Reasoning about data.

Z. Pawlak, Dordrecht, The Netherlands: Kluwer, 1991.

[68] Preparata, F. and Shamos, M. 1988. "Computational Geometry: an Introduction". Springer Verlag.

[69] A. Albanese and A. Petrosino, "A Non Parametric Approach to the Outlier Detection in Spatio-Temporal Data Analysis", Springer book "Information Technology and Innovation Trends in Organizations", D'Atri, et al., 2011.

[70] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD Int. Conf. on Management of Data, pages 427-438, Dallas, Texas, May 2000.

[71] G. Ratsch, S. Mika, B. Scholkopf and K.-R. MULLER, Constructing boosting algorithms from SVMS: An application to one-class classification. 2002. IEEE Trans. Patt. Anal. Mach. Intel. 24, 9, 1184-1199.

[72] Raymond T. Ng, Jiawei Han, "CLARANS: A Method for Clustering Objects for

Spatial Data Mining," IEEE Transactions on Knowledge and Data Engineering ", 2002

[73] D. Ren, I. Rahal, W. Perrizo, A Vertical Outlier Detection Algorithm with Clusters as By-product. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)

[74] S. Roberts and L. Tarassenko A probabilistic resource allocating network for novelty detection. 1994. Neural Comput. 6, 2, 270-284.

[75] S. Roberts, Novelty detection using extreme value statistics. 1999. In Proceedings of the IEEE Vision, Image and Signal Processing Conference Vol. 146. 124-129.

[76] S. Roberts, Extreme value statistics for novelty detection in biomedical signal processing. 2002. In Proceedings of the 1st International Conference on Advances in Medical Signal and Information Processing.166-172.

[77] D. M. Rocke and D. L. Woodruff, "Identification of Outliers in Multivariate Data," Journal of the American Statistical Association, vol.91, no. 435, pp. 1047-1061, 1996.

[78] B. Rosner, Percentage points for a generalized ESD many-outlier procedure. 1983. Technometrics 25, 2, 165-172.

[79] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection, New York, Wiley, 1987.

[80] P.J. Rousseeuw and A .M. Leroy (1996) Robust regression and outlier detection. John Wiley and Sons

[81] P. J. Rousseeuw and K. van Driessen, "A Fast Algorithm for the Minimum Covariance Determinant Estimator," Technometrics vol. 41, no. 3, pp. 212-223, 1999.

[82] I. Ruts and P. Rousseeuw (1996) Computing depth contours of bivariate point clouds. Journal of Computational Statistics and data Analysis, 23:153-168

[83] Coleman, J. E. Cantor, T. H. Neale, American National Government, "CRS Report for Congress", 2004

[84] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based Clustering in Spatial Databases: the algorithm GDBSCAN and its applications", Data Mining and Knowledge Discovery, vol.2, Jun. 1998, pp.169-194.

[85] Shekhar, S., Lu, C. T. and Zhang, P. (2001), Detecting graph-based spatial outliers: Algorithms and applications, in '7th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2001)', ACM Press, San Francisco, CA.

[86] Shewhart, W. A. 1931. Economic Control of Quality of Manufactured Product. D. Van Nostrand Company.

[87] Q. Song, W. Hu and W. Xie, Robust support vector machine with bullet hole image classification. 2002. IEEE Trans. Syst. Man Cyber. Part C: Applications and Reviews 32, 4.

[88] C. Spence, L. Parra, and P. Sajda, Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. 2001. In Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. IEEE Computer Society, 3.

[89] W. Stefansky, Rejecting outliers in factorial designs. 1972. Technometrics 14, 2, 469-479.

[90] C. Stefano, C. Sansone and M. Vento, To reject or not to reject: that is the question: An answer in the case of neural classifiers. 2000. IEEE Trans. Syst. Manag. Cyber. 30, 1, 84-94.

[91] P. Sun and S. Chawla, "On Local Spatial Outliers," Proc. 4-th IEEE Int'l Conf. on Data Mining, pp. 209-216, 2004.

[92] C. Surace and K. Worden A novelty detection method to diagnose damage in structures: An application to an offshore platform. 1998. In Proceedings of the 8th International Conference of Off-Shore and Polar Engineering. vol. 4. Colorado, 64-70.

[93] J. Tang, Z. Chen, A. W. Fu, F. W. Cheung (2002) Enhancing effectiveness of outlier detections for low density patterns. In: Proceedings of PAKDD, pp 535-548

[94] J. Tang, Z. Chen and A. W.-C. Fu, and D. Cheung, "A Robust Outlier Detection Scheme in Large Data Sets," Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining, LNCS 2336, pp. 535-548, 2002.

[95] L. Tarassenko, Novelty detection for the identification of masses in mammograms. 1995. In Proceedings of the 4th IEEE International Conference on Artificial Neural Networks. vol. 4. 442-447.

[96] D. Tax and R. Duin, Data domain description using support vectors. 1999a. In Proceedings of the European Symposium on Artificial Neural Networks, M. Verleysen, Ed., 251-256.

[97] D. Tax and R. Duin, Support vector data description. 1999b. Patt. Recog. Lett. 20, 11-13, 1191-1199.

[98] D. M. J. Tax, One-class classification; concept-learning in the absence of counter-examples. Ph.D. thesis, 2001. Delft University of Technology.

[99] R. S. Tsay, D. Pea and A. E. Pankratz, Outliers in multi-variate time series. 2000. Biometrika 87, 4, 789-804.

[100] J. Tukey (1997) Exploratory data analysis. Addison-Wesley

[101] P. M. Valero-Mora, F. W. Young, M. Friendly (2003) Visualizing categorical data in ViSta. Computational Statistics & Data Analysis, vol. 43, pp 495-508

[102] V. N. Vapnik, The Nature of Statistical Learning Theory. 1995. Springer-Verlag

[103] Sujing Wang, Nidal Zeidat, Christoph F. Eick, A 2D Spatial dataset Complex9.

[104] G. J. Williams and R. A. Baxter, H. He, S. Hawkins and L. Gu, A comparative study of RNN for outlier detection in data mining. 2002. In Proceedings of the IEEE International Conference on Data Mining. IEEE Computer Society, 709.

[105] E. Wu, W. Liu and S. Chawla. "Spatio-Temporal Outlier Detection in Precipitation Data". *In Proceedings of the second international Workshop on Knowledge Discovery From Sensor Data. SensorKDD'08*, LNCS 5840.

[106] K. Yamanishi and J. Ichi Takeuchi, Discovering outlier filtering rules from unla-

beled data: Combining a supervised learner with an unsupervised learner. 2001. In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, 389-394.

[107] K. Yamanishi, J. Ichi Takeuchi, G. Williams and P. Milne, Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. 2004. Data Min. Knowl. Disc. 8, 275-300.

[108] N. Ye and Q. Chen, An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. 2001. Quality Reliability Engin. Int. 17, 105-112.

[109] Yunxin Tao, Dechang Pi "Unifying Density-Based Clustering and Outlier Detection" *Second International Workshop on Knowledge Discovery and Data Mining* 2009

[110] T. Zhang, R. Ramakrishnan, M. Livny "BIRCH: An Efficient Data Clustering Method for Very Large Databases", ACM SIGMOD Conference Proceedings, 1996

[111] Yong Zhang, Su Yang, Yuanyuan Wang: LDBOD: A novel local distribution based outlier detector. Pattern Recognition Letters 29(7): 967-976 (2008)

[112] Ke Zhang, M. Hutter, W. Jin, "A New Local Distance-based Outlier Detection Ap-

proach for Scattered Real-World Data", Proc. 13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'09)

[113] Zhang Y, Meratnia N, Havinga P: A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets.Technical Report TR-CTIT-07-79, Centre for Telematics and Information Technology, University of Twente, Enschede 2007.

[114] S. Zhou, Y. Zhao, J. Guan, and J. Huang, "A Neighborhood-Based Clustering Algorithm", Proc. 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 05), Springer-Verlag, May. 2005, pp.361-371.